Abaqus 6.12

Analysis User's Manual

Volume III: Materials

Abaqus Analysis

User's Manual

Volume III

Legal Notices

CAUTION: This documentation is intended for qualified users who will exercise sound engineering judgment and expertise in the use of the Abaqus Software. The Abaqus Software is inherently complex, and the examples and procedures in this documentation are not intended to be exhaustive or to apply to any particular situation. Users are cautioned to satisfy themselves as to the accuracy and results of their analyses.

Dassault Systèmes and its subsidiaries, including Dassault Systèmes Simulia Corp., shall not be responsible for the accuracy or usefulness of any analysis performed using the Abaqus Software or the procedures, examples, or explanations in this documentation. Dassault Systèmes and its subsidiaries shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

The Abaqus Software is available only under license from Dassault Systèmes or its subsidiary and may be used or reproduced only in accordance with the terms of such license. This documentation is subject to the terms and conditions of either the software license agreement signed by the parties, or, absent such an agreement, the then current software license agreement to which the documentation relates.

This documentation and the software described in this documentation are subject to change without prior notice.

No part of this documentation may be reproduced or distributed in any form without prior written permission of Dassault Systèmes or its subsidiary.

The Abaqus Software is a product of Dassault Systèmes Simulia Corp., Providence, RI, USA.

© Dassault Systèmes, 2012

Abaqus, the 3DS logo, SIMULIA, CATIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of their respective owners. For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the Abaqus 6.12 Installation and Licensing Guide.

Locations

SIMULIA Worldwide Headquarters	Rising Sun Mills, 166 Valley Street, Providence, RI 02909–2499, Tel: +1 401 276 4400, Fax: +1 401 276 4408, simulia.support@3ds.com, http://www.simulia.com
SIMULIA European Headquarters	Stationsplein 8-K, 6221 BT Maastricht, The Netherlands, Tel: +31 43 7999 084, Fax: +31 43 7999 306, simulia.europe.info@3ds.com

Dassault Systèmes' Centers of Simulation Excellence

United States	Fremont, CA, Tel: +1 510 794 5891, simulia.west.support@3ds.com
	West Lafayette, IN, Tel: +1 765 497 1373, simulia.central.support@3ds.com
	Northville, MI, Tel: +1 248 349 4669, simulia.greatlakes.info@3ds.com
	Woodbury, MN, Tel: +1 612 424 9044, simulia.central.support@3ds.com
	Mayfield Heights, OH, Tel: +1 216 378 1070, simulia.erie.info@3ds.com
	Mason, OH, Tel: +1 513 275 1430, simulia.central.support@3ds.com
	Warwick, RI, Tel: +1 401 739 3637, simulia.east.support@3ds.com
	Lewisville, TX, Tel: +1 972 221 6500, simulia.south.info@3ds.com
Australia	Richmond VIC, Tel: +61 3 9421 2900, simulia.au.support@3ds.com
Austria	Vienna, Tel: +43 1 22 707 200, simulia.at.info@3ds.com
Benelux	Maarssen, The Netherlands, Tel: +31 346 585 710, simulia.benelux.support@3ds.com
Canada	Toronto, ON, Tel: +1 416 402 2219, simulia.greatlakes.info@3ds.com
China	Beijing, P. R. China, Tel: +8610 6536 2288, simulia.cn.support@3ds.com
	Shanghai, P. R. China, Tel: +8621 3856 8000, simulia.cn.support@3ds.com
Finland	Espoo, Tel: +358 40 902 2973, simulia.nordic.info@3ds.com
France	Velizy Villacoublay Cedex, Tel: +33 1 61 62 72 72, simulia.fr.support@3ds.com
Germany	Aachen, Tel: +49 241 474 01 0, simulia.de.info@3ds.com
	Munich, Tel: +49 89 543 48 77 0, simulia.de.info@3ds.com
India	Chennai, Tamil Nadu, Tel: +91 44 43443000, simulia.in.info@3ds.com
Italy	Lainate MI, Tel: +39 02 3343061, simulia.ity.info@3ds.com
Japan	Tokyo, Tel: +81 3 5442 6302, simulia.jp.support@3ds.com
	Osaka, Tel: +81 6 7730 2703, simulia.jp.support@3ds.com
Korea	Mapo-Gu, Seoul, Tel: +82 2 785 6707/8, simulia.kr.info@3ds.com
Latin America	Puerto Madero, Buenos Aires, Tel: +54 11 4312 8700, Horacio.Burbridge@3ds.com
Scandinavia	Stockholm, Sweden, Tel: +46 8 68430450, simulia.nordic.info@3ds.com
United Kingdom	Warrington, Tel: +44 1925 830900, simulia.uk.info@3ds.com

Authorized Support Centers

Argentina	SMARTtech Sudamerica SRL, Buenos Aires, Tel: +54 11 4717 2717
	KB Engineering, Buenos Aires, Tel: +54 11 4326 7542
	Solaer Ingeniería, Buenos Aires, Tel: +54 221 489 1738
Brazil	SMARTtech Mecânica, Sao Paulo-SP, Tel: +55 11 3168 3388
Czech & Slovak Republics	Synerma s. r. o., Psáry, Prague-West, Tel: +420 603 145 769, abaqus@synerma.cz
Greece	3 Dimensional Data Systems, Crete, Tel: +30 2821040012, support@3dds.gr
Israel	ADCOM, Givataim, Tel: +972 3 7325311, shmulik.keidar@adcomsim.co.il
Malaysia	WorleyParsons Services Sdn. Bhd., Kuala Lumpur, Tel: +603 2039 9000, abaqus.my@worleyparsons.com
Mexico	Kimeca.NET SA de CV, Mexico, Tel: +52 55 2459 2635
New Zealand	Matrix Applied Computing Ltd., Auckland, Tel: +64 9 623 1223, abaqus-tech@matrix.co.nz
Poland	BudSoft Sp. z o.o., Poznań, Tel: +48 61 8508 466, info@budsoft.com.pl
Russia, Belarus & Ukraine	TESIS Ltd., Moscow, Tel: +7 495 612 44 22, info@tesis.com.ru
Singapore	WorleyParsons Pte Ltd., Singapore, Tel: +65 6735 8444, abaqus.sg@worleyparsons.com
South Africa	Finite Element Analysis Services (Pty) Ltd., Parklands, Tel: +27 21 556 6462, feas@feas.co.za
Spain & Portugal	Principia Ingenieros Consultores, S.A., Madrid, Tel: +34 91 209 1482, simulia@principia.es

Taiwan	Simutech Solution Corporation, Taipei, R.O.C., Tel: +886 2 2507 9550, camilla@simutech.com.tw
Thailand	WorleyParsons Pte Ltd., Singapore, Tel: +65 6735 8444, abaqus.sg@worleyparsons.com
Turkey	A-Ztech Ltd., Istanbul, Tel: +90 216 361 8850, info@a-ztech.com.tr

Complete contact information is available at http://www.simulia.com/locations/locations.html.

Preface

This section lists various resources that are available for help with using Abaqus Unified FEA software.

Support

Both technical engineering support (for problems with creating a model or performing an analysis) and systems support (for installation, licensing, and hardware-related problems) for Abaqus are offered through a network of local support offices. Regional contact information is listed in the front of each Abaqus manual and is accessible from the **Locations** page at www.simulia.com.

Support for SIMULIA products

SIMULIA provides a knowledge database of answers and solutions to questions that we have answered, as well as guidelines on how to use Abaqus, SIMULIA Scenario Definition, Isight, and other SIMULIA products. You can also submit new requests for support. All support incidents are tracked. If you contact us by means outside the system to discuss an existing support problem and you know the incident or support request number, please mention it so that we can query the database to see what the latest action has been.

Many questions about Abaqus can also be answered by visiting the **Products** page and the **Support** page at www.simulia.com.

Anonymous ftp site

To facilitate data transfer with SIMULIA, an anonymous ftp account is available at **`ftp.simulia.com`**. Login as user **`anonymous`**, and type your e-mail address as your password. Contact support before placing files on the site.

Training

All offices and representatives offer regularly scheduled public training classes. The courses are offered in a traditional classroom form and via the Web. We also provide training seminars at customer sites. All training classes and seminars include workshops to provide as much practical experience with Abaqus as possible. For a schedule and descriptions of available classes, see www.simulia.com or call your local office or representative.

Feedback

We welcome any suggestions for improvements to Abaqus software, the support program, or documentation. We will ensure that any enhancement requests you make are considered for future releases. If you wish to make a suggestion about the service or products, refer to www.simulia.com. Complaints should be made by contacting your local office or through www.simulia.com by visiting the **Quality Assurance** section of the **Support** page.

Contents

Volume I

PART I INTRODUCTION, SPATIAL MODELING, AND EXECUTION

1. Introduction

Introduction: general 1.1.1

Abaqus syntax and conventions

Input syntax rules 1.2.1
Conventions 1.2.2

Abaqus model definition

Defining a model in Abaqus 1.3.1

Parametric modeling

Parametric input 1.4.1

2. Spatial Modeling

Node definition

Node definition 2.1.1
Parametric shape variation 2.1.2
Nodal thicknesses 2.1.3
Normal definitions at nodes 2.1.4
Transformed coordinate systems 2.1.5
Adjusting nodal coordinates 2.1.6

Element definition

Element definition 2.2.1
Element foundations 2.2.2
Defining reinforcement 2.2.3
Defining rebar as an element property 2.2.4
Orientations 2.2.5

Surface definition

Surfaces: overview 2.3.1
Element-based surface definition 2.3.2
Node-based surface definition 2.3.3
Analytical rigid surface definition 2.3.4

Eulerian surface definition 2.3.5
Operating on surfaces 2.3.6

Rigid body definition

Rigid body definition 2.4.1

Integrated output section definition

Integrated output section definition 2.5.1

Mass adjustment

Adjust and/or redistribute mass of an element set 2.6.1

Nonstructural mass definition

Nonstructural mass definition 2.7.1

Distribution definition

Distribution definition 2.8.1

Display body definition

Display body definition 2.9.1

Assembly definition

Defining an assembly 2.10.1

Matrix definition

Defining matrices 2.11.1

3. Job Execution

Execution procedures: overview

Execution procedure for Abaqus: overview 3.1.1

Execution procedures

Obtaining information 3.2.1
Abaqus/Standard, Abaqus/Explicit, and Abaqus/CFD execution 3.2.2
SIMULIA Co-Simulation Engine controller execution 3.2.3
Abaqus/Standard, Abaqus/Explicit, and Abaqus/CFD co-simulation execution 3.2.4
Abaqus/CAE execution 3.2.5
Abaqus/Viewer execution 3.2.6
Python execution 3.2.7
Parametric studies 3.2.8
Abaqus documentation 3.2.9
Licensing utilities 3.2.10
ASCII translation of results (`.fil`) files 3.2.11
Joining results (`.fil`) files 3.2.12
Querying the keyword/problem database 3.2.13

Fetching sample input files 3.2.14
Making user-defined executables and subroutines 3.2.15
Input file and output database upgrade utility 3.2.16
Generating output database reports 3.2.17
Joining output database (`.odb`) files from restarted analyses 3.2.18
Combining output from substructures 3.2.19
Combining data from multiple output databases 3.2.20
Network output database file connector 3.2.21
Mapping thermal and magnetic loads 3.2.22
Fixed format conversion utility 3.2.23
Translating Nastran bulk data files to Abaqus input files 3.2.24
Translating Abaqus files to Nastran bulk data files 3.2.25
Translating ANSYS input files to Abaqus input files 3.2.26
Translating PAM-CRASH input files to partial Abaqus input files 3.2.27
Translating RADIOSS input files to partial Abaqus input files 3.2.28
Translating Abaqus output database files to Nastran Output2 results files 3.2.29
Translating LS-DYNA data files to Abaqus input files 3.2.30
Exchanging Abaqus data with ZAERO 3.2.31
Encrypting and decrypting Abaqus input data 3.2.32
Job execution control 3.2.33

Environment file settings

Using the Abaqus environment settings 3.3.1

Managing memory and disk resources

Managing memory and disk use in Abaqus 3.4.1

Parallel execution

Parallel execution: overview 3.5.1
Parallel execution in Abaqus/Standard 3.5.2
Parallel execution in Abaqus/Explicit 3.5.3
Parallel execution in Abaqus/CFD 3.5.4

File extension definitions

File extensions used by Abaqus 3.6.1

FORTRAN unit numbers

FORTRAN unit numbers used by Abaqus 3.7.1

PART II OUTPUT

4. Output

Output 4.1.1
Output to the data and results files 4.1.2
Output to the output database 4.1.3
Error indicator output 4.1.4

Output variables

Abaqus/Standard output variable identifiers 4.2.1
Abaqus/Explicit output variable identifiers 4.2.2
Abaqus/CFD output variable identifiers 4.2.3

The postprocessing calculator

The postprocessing calculator 4.3.1

5. File Output Format

Accessing the results file

Accessing the results file: overview 5.1.1
Results file output format 5.1.2
Accessing the results file information 5.1.3
Utility routines for accessing the results file 5.1.4

OI.1 Abaqus/Standard Output Variable Index

OI.2 Abaqus/Explicit Output Variable Index

OI.3 Abaqus/CFD Output Variable Index

Volume II

PART III ANALYSIS PROCEDURES, SOLUTION, AND CONTROL

6. Analysis Procedures

Introduction

Solving analysis problems: overview 6.1.1
Defining an analysis 6.1.2
General and linear perturbation procedures 6.1.3
Multiple load case analysis 6.1.4
Direct linear equation solver 6.1.5
Iterative linear equation solver 6.1.6

Static stress/displacement analysis

Static stress analysis procedures: overview 6.2.1
Static stress analysis 6.2.2
Eigenvalue buckling prediction 6.2.3
Unstable collapse and postbuckling analysis 6.2.4
Quasi-static analysis 6.2.5
Direct cyclic analysis 6.2.6
Low-cycle fatigue analysis using the direct cyclic approach 6.2.7

Dynamic stress/displacement analysis

Dynamic analysis procedures: overview 6.3.1
Implicit dynamic analysis using direct integration 6.3.2
Explicit dynamic analysis 6.3.3
Direct-solution steady-state dynamic analysis 6.3.4
Natural frequency extraction 6.3.5
Complex eigenvalue extraction 6.3.6
Transient modal dynamic analysis 6.3.7
Mode-based steady-state dynamic analysis 6.3.8
Subspace-based steady-state dynamic analysis 6.3.9
Response spectrum analysis 6.3.10
Random response analysis 6.3.11

Steady-state transport analysis

Steady-state transport analysis 6.4.1

Heat transfer and thermal-stress analysis

Heat transfer analysis procedures: overview 6.5.1
Uncoupled heat transfer analysis 6.5.2

Fully coupled thermal-stress analysis 6.5.3
Adiabatic analysis 6.5.4

Fluid dynamic analysis

Fluid dynamic analysis procedures: overview 6.6.1
Incompressible fluid dynamic analysis 6.6.2

Electromagnetic analysis

Electromagnetic analysis procedures 6.7.1
Piezoelectric analysis 6.7.2
Coupled thermal-electrical analysis 6.7.3
Fully coupled thermal-electrical-structural analysis 6.7.4
Eddy current analysis 6.7.5
Magnetostatic analysis 6.7.6

Coupled pore fluid flow and stress analysis

Coupled pore fluid diffusion and stress analysis 6.8.1
Geostatic stress state 6.8.2

Mass diffusion analysis

Mass diffusion analysis 6.9.1

Acoustic and shock analysis

Acoustic, shock, and coupled acoustic-structural analysis 6.10.1

Abaqus/Aqua analysis

Abaqus/Aqua analysis 6.11.1

Annealing

Annealing procedure 6.12.1

7. Analysis Solution and Control

Solving nonlinear problems

Solving nonlinear problems 7.1.1

Analysis convergence controls

Convergence and time integration criteria: overview 7.2.1
Commonly used control parameters 7.2.2
Convergence criteria for nonlinear problems 7.2.3
Time integration accuracy in transient problems 7.2.4

PART IV ANALYSIS TECHNIQUES

8. Analysis Techniques: Introduction

Analysis techniques: overview 8.1.1

9. Analysis Continuation Techniques

Restarting an analysis

Restarting an analysis 9.1.1

Importing and transferring results

Transferring results between Abaqus analyses: overview 9.2.1
Transferring results between Abaqus/Explicit and Abaqus/Standard 9.2.2
Transferring results from one Abaqus/Standard analysis to another 9.2.3
Transferring results from one Abaqus/Explicit analysis to another 9.2.4

10. Modeling Abstractions

Substructuring

Using substructures 10.1.1
Defining substructures 10.1.2

Submodeling

Submodeling: overview 10.2.1
Node-based submodeling 10.2.2
Surface-based submodeling 10.2.3

Generating global matrices

Generating matrices 10.3.1

Symmetric model generation, results transfer, and analysis of cyclic symmetry models

Symmetric model generation 10.4.1
Transferring results from a symmetric mesh or a partial three-dimensional mesh to a full three-dimensional mesh 10.4.2
Analysis of models that exhibit cyclic symmetry 10.4.3

Periodic media analysis

Periodic media analysis 10.5.1

Meshed beam cross-sections

Meshed beam cross-sections 10.6.1

Modeling discontinuities as an enriched feature using the extended finite element method

Modeling discontinuities as an enriched feature using the extended finite element method 10.7.1

11. Special-Purpose Techniques

Inertia relief

Inertia relief 11.1.1

Mesh modification or replacement

Element and contact pair removal and reactivation 11.2.1

Geometric imperfections

Introducing a geometric imperfection into a model 11.3.1

Fracture mechanics

Fracture mechanics: overview 11.4.1
Contour integral evaluation 11.4.2
Crack propagation analysis 11.4.3

Surface-based fluid modeling

Surface-based fluid cavities: overview 11.5.1
Fluid cavity definition 11.5.2
Fluid exchange definition 11.5.3
Inflator definition 11.5.4

Mass scaling

Mass scaling 11.6.1

Selective subcycling

Selective subcycling 11.7.1

Steady-state detection

Steady-state detection 11.8.1

12. Adaptivity Techniques

Adaptivity techniques: overview

Adaptivity techniques 12.1.1

ALE adaptive meshing

ALE adaptive meshing: overview 12.2.1
Defining ALE adaptive mesh domains in Abaqus/Explicit 12.2.2
ALE adaptive meshing and remapping in Abaqus/Explicit 12.2.3
Modeling techniques for Eulerian adaptive mesh domains in Abaqus/Explicit 12.2.4

Output and diagnostics for ALE adaptive meshing in Abaqus/Explicit 12.2.5
Defining ALE adaptive mesh domains in Abaqus/Standard 12.2.6
ALE adaptive meshing and remapping in Abaqus/Standard 12.2.7

Adaptive remeshing

Adaptive remeshing: overview 12.3.1
Selection of error indicators influencing adaptive remeshing 12.3.2
Solution-based mesh sizing 12.3.3

Analysis continuation after mesh replacement

Mesh-to-mesh solution mapping 12.4.1

13. Optimization Techniques

Structural optimization: overview

Structural optimization: overview 13.1.1

Optimization models

Design responses 13.2.1
Objectives and constraints 13.2.2
Creating Abaqus optimization models 13.2.3

14. Eulerian Analysis

Eulerian analysis 14.1.1
Defining Eulerian boundaries 14.1.2
Eulerian mesh motion 14.1.3
Defining adaptive mesh refinement in the Eulerian domain 14.1.4

15. Particle Methods

Smoothed particle hydrodynamic analyses

Smoothed particle hydrodynamic analysis 15.1.1
Finite element conversion to SPH particles 15.1.2

16. Sequentially Coupled Multiphysics Analyses

Predefined fields for sequential coupling 16.1.1
Sequentially coupled thermal-stress analysis 16.1.2
Predefined loads for sequential coupling 16.1.3

17. Co-simulation

Co-simulation: overview 17.1.1

Preparing an Abaqus analysis for co-simulation

Preparing an Abaqus analysis for co-simulation 17.2.1

Co-simulation between Abaqus solvers

Abaqus/Standard to Abaqus/Explicit co-simulation 17.3.1
Abaqus/CFD to Abaqus/Standard or to Abaqus/Explicit co-simulation 17.3.2

18. Extending Abaqus Analysis Functionality

User subroutines and utilities

User subroutines: overview 18.1.1
Available user subroutines 18.1.2
Available utility routines 18.1.3

19. Design Sensitivity Analysis

Design sensitivity analysis 19.1.1

20. Parametric Studies

Scripting parametric studies

Scripting parametric studies 20.1.1

Parametric studies: commands

aStudy.combine(): Combine parameter samples for parametric studies. 20.2.1
aStudy.constrain(): Constrain parameter value combinations in parametric studies. 20.2.2
aStudy.define(): Define parameters for parametric studies. 20.2.3
aStudy.execute(): Execute the analysis of parametric study designs. 20.2.4
aStudy.gather(): Gather the results of a parametric study. 20.2.5
aStudy.generate(): Generate the analysis job data for a parametric study. 20.2.6
aStudy.output(): Specify the source of parametric study results. 20.2.7
aStudy=ParStudy(): Create a parametric study. 20.2.8
aStudy.report(): Report parametric study results. 20.2.9
aStudy.sample(): Sample parameters for parametric studies. 20.2.10

Volume III

PART V MATERIALS

21. Materials: Introduction

Introduction

Material library: overview 21.1.1
Material data definition 21.1.2
Combining material behaviors 21.1.3

General properties

Density 21.2.1

22. Elastic Mechanical Properties

Overview

Elastic behavior: overview 22.1.1

Linear elasticity

Linear elastic behavior 22.2.1
No compression or no tension 22.2.2
Plane stress orthotropic failure measures 22.2.3

Porous elasticity

Elastic behavior of porous materials 22.3.1

Hypoelasticity

Hypoelastic behavior 22.4.1

Hyperelasticity

Hyperelastic behavior of rubberlike materials 22.5.1
Hyperelastic behavior in elastomeric foams 22.5.2
Anisotropic hyperelastic behavior 22.5.3

Stress softening in elastomers

Mullins effect 22.6.1
Energy dissipation in elastomeric foams 22.6.2

Viscoelasticity

Time domain viscoelasticity 22.7.1
Frequency domain viscoelasticity 22.7.2

Nonlinear viscoelasticity

Hysteresis in elastomers 22.8.1
Parallel network viscoelastic model 22.8.2

Rate sensitive elastomeric foams

Low-density foams 22.9.1

23. Inelastic Mechanical Properties

Overview

Inelastic behavior 23.1.1

Metal plasticity

Classical metal plasticity 23.2.1
Models for metals subjected to cyclic loading 23.2.2
Rate-dependent yield 23.2.3
Rate-dependent plasticity: creep and swelling 23.2.4
Annealing or melting 23.2.5
Anisotropic yield/creep 23.2.6
Johnson-Cook plasticity 23.2.7
Dynamic failure models 23.2.8
Porous metal plasticity 23.2.9
Cast iron plasticity 23.2.10
Two-layer viscoplasticity 23.2.11
ORNL – Oak Ridge National Laboratory constitutive model 23.2.12
Deformation plasticity 23.2.13

Other plasticity models

Extended Drucker-Prager models 23.3.1
Modified Drucker-Prager/Cap model 23.3.2
Mohr-Coulomb plasticity 23.3.3
Critical state (clay) plasticity model 23.3.4
Crushable foam plasticity models 23.3.5

Fabric materials

Fabric material behavior 23.4.1

Jointed materials

Jointed material model 23.5.1

Concrete

Concrete smeared cracking 23.6.1
Cracking model for concrete 23.6.2
Concrete damaged plasticity 23.6.3

Permanent set in rubberlike materials

Permanent set in rubberlike materials 23.7.1

24. Progressive Damage and Failure

Progressive damage and failure: overview

Progressive damage and failure 24.1.1

Damage and failure for ductile metals

Damage and failure for ductile metals: overview 24.2.1
Damage initiation for ductile metals 24.2.2
Damage evolution and element removal for ductile metals 24.2.3

Damage and failure for fiber-reinforced composites

Damage and failure for fiber-reinforced composites: overview 24.3.1
Damage initiation for fiber-reinforced composites 24.3.2
Damage evolution and element removal for fiber-reinforced composites 24.3.3

Damage and failure for ductile materials in low-cycle fatigue analysis

Damage and failure for ductile materials in low-cycle fatigue analysis: overview 24.4.1
Damage initiation for ductile materials in low-cycle fatigue 24.4.2
Damage evolution for ductile materials in low-cycle fatigue 24.4.3

25. Hydrodynamic Properties

Overview

Hydrodynamic behavior: overview 25.1.1

Equations of state

Equation of state 25.2.1

26. Other Material Properties

Mechanical properties

Material damping 26.1.1
Thermal expansion 26.1.2
Field expansion 26.1.3
Viscosity 26.1.4

Heat transfer properties

Thermal properties: overview 26.2.1
Conductivity 26.2.2
Specific heat 26.2.3
Latent heat 26.2.4

Acoustic properties

Acoustic medium 26.3.1

Mass diffusion properties

Diffusivity 26.4.1
Solubility 26.4.2

Electromagnetic properties

Electrical conductivity 26.5.1
Piezoelectric behavior 26.5.2
Magnetic permeability 26.5.3

Pore fluid flow properties

Pore fluid flow properties 26.6.1
Permeability 26.6.2
Porous bulk moduli 26.6.3
Sorption 26.6.4
Swelling gel 26.6.5
Moisture swelling 26.6.6

User materials

User-defined mechanical material behavior 26.7.1
User-defined thermal material behavior 26.7.2

Volume IV

PART VI ELEMENTS

27. Elements: Introduction

Element library: overview 27.1.1
Choosing the element's dimensionality 27.1.2
Choosing the appropriate element for an analysis type 27.1.3
Section controls 27.1.4

28. Continuum Elements

General-purpose continuum elements

Solid (continuum) elements 28.1.1
One-dimensional solid (link) element library 28.1.2
Two-dimensional solid element library 28.1.3
Three-dimensional solid element library 28.1.4
Cylindrical solid element library 28.1.5
Axisymmetric solid element library 28.1.6
Axisymmetric solid elements with nonlinear, asymmetric deformation 28.1.7

Fluid continuum elements

Fluid (continuum) elements 28.2.1
Fluid element library 28.2.2

Infinite elements

Infinite elements 28.3.1
Infinite element library 28.3.2

Warping elements

Warping elements 28.4.1
Warping element library 28.4.2

Particle elements

Particle elements 28.5.1
Particle element library 28.5.2

29. Structural Elements

Membrane elements

Membrane elements 29.1.1
General membrane element library 29.1.2
Cylindrical membrane element library 29.1.3

Axisymmetric membrane element library 29.1.4

Truss elements

Truss elements 29.2.1
Truss element library 29.2.2

Beam elements

Beam modeling: overview 29.3.1
Choosing a beam cross-section 29.3.2
Choosing a beam element 29.3.3
Beam element cross-section orientation 29.3.4
Beam section behavior 29.3.5
Using a beam section integrated during the analysis to define the section behavior 29.3.6
Using a general beam section to define the section behavior 29.3.7
Beam element library 29.3.8
Beam cross-section library 29.3.9

Frame elements

Frame elements 29.4.1
Frame section behavior 29.4.2
Frame element library 29.4.3

Elbow elements

Pipes and pipebends with deforming cross-sections: elbow elements 29.5.1
Elbow element library 29.5.2

Shell elements

Shell elements: overview 29.6.1
Choosing a shell element 29.6.2
Defining the initial geometry of conventional shell elements 29.6.3
Shell section behavior 29.6.4
Using a shell section integrated during the analysis to define the section behavior 29.6.5
Using a general shell section to define the section behavior 29.6.6
Three-dimensional conventional shell element library 29.6.7
Continuum shell element library 29.6.8
Axisymmetric shell element library 29.6.9
Axisymmetric shell elements with nonlinear, asymmetric deformation 29.6.10

30. Inertial, Rigid, and Capacitance Elements

Point mass elements

Point masses 30.1.1
Mass element library 30.1.2

Rotary inertia elements

Rotary inertia 30.2.1
Rotary inertia element library 30.2.2

Rigid elements

Rigid elements 30.3.1
Rigid element library 30.3.2

Capacitance elements

Point capacitance 30.4.1
Capacitance element library 30.4.2

31. Connector Elements

Connector elements

Connectors: overview 31.1.1
Connector elements 31.1.2
Connector actuation 31.1.3
Connector element library 31.1.4
Connection-type library 31.1.5

Connector element behavior

Connector behavior 31.2.1
Connector elastic behavior 31.2.2
Connector damping behavior 31.2.3
Connector functions for coupled behavior 31.2.4
Connector friction behavior 31.2.5
Connector plastic behavior 31.2.6
Connector damage behavior 31.2.7
Connector stops and locks 31.2.8
Connector failure behavior 31.2.9
Connector uniaxial behavior 31.2.10

32. Special-Purpose Elements

Spring elements

Springs 32.1.1
Spring element library 32.1.2

Dashpot elements

Dashpots 32.2.1
Dashpot element library 32.2.2

Flexible joint elements

Flexible joint element 32.3.1
Flexible joint element library 32.3.2

Distributing coupling elements

Distributing coupling elements 32.4.1
Distributing coupling element library 32.4.2

Cohesive elements

Cohesive elements: overview 32.5.1
Choosing a cohesive element 32.5.2
Modeling with cohesive elements 32.5.3
Defining the cohesive element's initial geometry 32.5.4
Defining the constitutive response of cohesive elements using a continuum approach 32.5.5
Defining the constitutive response of cohesive elements using a traction-separation description 32.5.6
Defining the constitutive response of fluid within the cohesive element gap 32.5.7
Two-dimensional cohesive element library 32.5.8
Three-dimensional cohesive element library 32.5.9
Axisymmetric cohesive element library 32.5.10

Gasket elements

Gasket elements: overview 32.6.1
Choosing a gasket element 32.6.2
Including gasket elements in a model 32.6.3
Defining the gasket element's initial geometry 32.6.4
Defining the gasket behavior using a material model 32.6.5
Defining the gasket behavior directly using a gasket behavior model 32.6.6
Two-dimensional gasket element library 32.6.7
Three-dimensional gasket element library 32.6.8
Axisymmetric gasket element library 32.6.9

Surface elements

Surface elements 32.7.1
General surface element library 32.7.2
Cylindrical surface element library 32.7.3
Axisymmetric surface element library 32.7.4

Tube support elements

Tube support elements 32.8.1
Tube support element library 32.8.2

Line spring elements

Line spring elements for modeling part-through cracks in shells 32.9.1
Line spring element library 32.9.2

Elastic-plastic joints

Elastic-plastic joints 32.10.1
Elastic-plastic joint element library 32.10.2

Drag chain elements

Drag chains 32.11.1
Drag chain element library 32.11.2

Pipe-soil elements

Pipe-soil interaction elements 32.12.1
Pipe-soil interaction element library 32.12.2

Acoustic interface elements

Acoustic interface elements 32.13.1
Acoustic interface element library 32.13.2

Eulerian elements

Eulerian elements 32.14.1
Eulerian element library 32.14.2

User-defined elements

User-defined elements 32.15.1
User-defined element library 32.15.2

EI.1 Abaqus/Standard Element Index

EI.2 Abaqus/Explicit Element Index

EI.3 Abaqus/CFD Element Index

Volume V

PART VII PRESCRIBED CONDITIONS

33. Prescribed Conditions

Overview

Prescribed conditions: overview 33.1.1
Amplitude curves 33.1.2

Initial conditions

Initial conditions in Abaqus/Standard and Abaqus/Explicit 33.2.1
Initial conditions in Abaqus/CFD 33.2.2

Boundary conditions

Boundary conditions in Abaqus/Standard and Abaqus/Explicit 33.3.1
Boundary conditions in Abaqus/CFD 33.3.2

Loads

Applying loads: overview 33.4.1
Concentrated loads 33.4.2
Distributed loads 33.4.3
Thermal loads 33.4.4
Electromagnetic loads 33.4.5
Acoustic and shock loads 33.4.6
Pore fluid flow 33.4.7

Prescribed assembly loads

Prescribed assembly loads 33.5.1

Predefined fields

Predefined fields 33.6.1

PART VIII CONSTRAINTS

34. Constraints

Overview

Kinematic constraints: overview 34.1.1

Multi-point constraints

Linear constraint equations 34.2.1

General multi-point constraints 34.2.2
Kinematic coupling constraints 34.2.3

Surface-based constraints

Mesh tie constraints 34.3.1
Coupling constraints 34.3.2
Shell-to-solid coupling 34.3.3
Mesh-independent fasteners 34.3.4

Embedded elements

Embedded elements 34.4.1

Element end release

Element end release 34.5.1

Overconstraint checks

Overconstraint checks 34.6.1

PART IX INTERACTIONS

35. Defining Contact Interactions

Overview

Contact interaction analysis: overview 35.1.1

Defining general contact in Abaqus/Standard

Defining general contact interactions in Abaqus/Standard 35.2.1
Surface properties for general contact in Abaqus/Standard 35.2.2
Contact properties for general contact in Abaqus/Standard 35.2.3
Controlling initial contact status in Abaqus/Standard 35.2.4
Stabilization for general contact in Abaqus/Standard 35.2.5
Numerical controls for general contact in Abaqus/Standard 35.2.6

Defining contact pairs in Abaqus/Standard

Defining contact pairs in Abaqus/Standard 35.3.1
Assigning surface properties for contact pairs in Abaqus/Standard 35.3.2
Assigning contact properties for contact pairs in Abaqus/Standard 35.3.3
Modeling contact interference fits in Abaqus/Standard 35.3.4
Adjusting initial surface positions and specifying initial clearances in Abaqus/Standard contact pairs 35.3.5
Adjusting contact controls in Abaqus/Standard 35.3.6
Defining tied contact in Abaqus/Standard 35.3.7
Extending master surfaces and slide lines 35.3.8

Contact modeling if substructures are present 35.3.9
Contact modeling if asymmetric-axisymmetric elements are present 35.3.10

Defining general contact in Abaqus/Explicit

Defining general contact interactions in Abaqus/Explicit 35.4.1
Assigning surface properties for general contact in Abaqus/Explicit 35.4.2
Assigning contact properties for general contact in Abaqus/Explicit 35.4.3
Controlling initial contact status for general contact in Abaqus/Explicit 35.4.4
Contact controls for general contact in Abaqus/Explicit 35.4.5

Defining contact pairs in Abaqus/Explicit

Defining contact pairs in Abaqus/Explicit 35.5.1
Assigning surface properties for contact pairs in Abaqus/Explicit 35.5.2
Assigning contact properties for contact pairs in Abaqus/Explicit 35.5.3
Adjusting initial surface positions and specifying initial clearances for contact pairs in Abaqus/Explicit 35.5.4
Contact controls for contact pairs in Abaqus/Explicit 35.5.5

36. Contact Property Models

Mechanical contact properties

Mechanical contact properties: overview 36.1.1
Contact pressure-overclosure relationships 36.1.2
Contact damping 36.1.3
Contact blockage 36.1.4
Frictional behavior 36.1.5
User-defined interfacial constitutive behavior 36.1.6
Pressure penetration loading 36.1.7
Interaction of debonded surfaces 36.1.8
Breakable bonds 36.1.9
Surface-based cohesive behavior 36.1.10

Thermal contact properties

Thermal contact properties 36.2.1

Electrical contact properties

Electrical contact properties 36.3.1

Pore fluid contact properties

Pore fluid contact properties 36.4.1

37. Contact Formulations and Numerical Methods

Contact formulations and numerical methods in Abaqus/Standard

Contact formulations in Abaqus/Standard 37.1.1

Contact constraint enforcement methods in Abaqus/Standard 37.1.2
Smoothing contact surfaces in Abaqus/Standard 37.1.3

Contact formulations and numerical methods in Abaqus/Explicit

Contact formulation for general contact in Abaqus/Explicit 37.2.1
Contact formulations for contact pairs in Abaqus/Explicit 37.2.2
Contact constraint enforcement methods in Abaqus/Explicit 37.2.3

38. Contact Difficulties and Diagnostics

Resolving contact difficulties in Abaqus/Standard

Contact diagnostics in an Abaqus/Standard analysis 38.1.1
Common difficulties associated with contact modeling in Abaqus/Standard 38.1.2

Resolving contact difficulties in Abaqus/Explicit

Contact diagnostics in an Abaqus/Explicit analysis 38.2.1
Common difficulties associated with contact modeling using contact pairs in Abaqus/Explicit 38.2.2

39. Contact Elements in Abaqus/Standard

Contact modeling with elements

Contact modeling with elements 39.1.1

Gap contact elements

Gap contact elements 39.2.1
Gap element library 39.2.2

Tube-to-tube contact elements

Tube-to-tube contact elements 39.3.1
Tube-to-tube contact element library 39.3.2

Slide line contact elements

Slide line contact elements 39.4.1
Axisymmetric slide line element library 39.4.2

Rigid surface contact elements

Rigid surface contact elements 39.5.1
Axisymmetric rigid surface contact element library 39.5.2

40. Defining Cavity Radiation in Abaqus/Standard

Cavity radiation 40.1.1

Part V: Materials

- Chapter 21, “Materials: Introduction”
- Chapter 22, “Elastic Mechanical Properties”
- Chapter 23, “Inelastic Mechanical Properties”
- Chapter 24, “Progressive Damage and Failure”
- Chapter 25, “Hydrodynamic Properties”
- Chapter 26, “Other Material Properties”

21. Materials: Introduction

Introduction 21.1
General properties 21.2

21.1 Introduction

- "Material library: overview," Section 21.1.1
- "Material data definition," Section 21.1.2
- "Combining material behaviors," Section 21.1.3

21.1.1 MATERIAL LIBRARY: OVERVIEW

This chapter describes how to define materials in Abaqus and contains brief descriptions of each of the material behaviors provided. Further details of the more advanced behaviors are provided in the Abaqus Theory Manual.

Defining materials

Materials are defined by:

- selecting material behaviors and defining them ("Material data definition," Section 21.1.2); and
- combining complementary material behaviors such as elasticity and plasticity ("Combining material behaviors," Section 21.1.3).

A local coordinate system can be used for material calculations ("Orientations," Section 2.2.5). Any anisotropic properties must be given in this local system.

Available material behaviors

The material library in Abaqus is intended to provide comprehensive coverage of both linear and nonlinear, isotropic and anisotropic material behaviors. The use of numerical integration in the elements, including numerical integration across the cross-sections of shells and beams, provides the flexibility to analyze the most complex composite structures.

Material behaviors fall into the following general categories:

- general properties (material damping, density, thermal expansion);
- elastic mechanical properties;
- inelastic mechanical properties;
- thermal properties;
- acoustic properties;
- hydrostatic fluid properties;
- equations of state;
- mass diffusion properties;
- electrical properties; and
- pore fluid flow properties.

Some of the mechanical behaviors offered are mutually exclusive: such behaviors cannot appear together in a single material definition. Some behaviors require the presence of other behaviors; for example, plasticity requires linear elasticity. Such requirements are discussed at the end of each material behavior description, as well as in "Combining material behaviors," Section 21.1.3.

Using material behaviors with various element types

There are no general restrictions on the use of particular material behaviors with solid, shell, beam, and pipe elements. Any combination that makes sense is acceptable. The few restrictions that do exist are mentioned when that particular behavior is described in the pages that follow. A section on the elements available for use with a material behavior appears at the end of each material behavior description.

Using complete material definitions

A material definition can include behaviors that are not meaningful for the elements or analysis in which the material is being used. Such behaviors will be ignored. For example, a material definition can include heat transfer properties (conductivity, specific heat) as well as stress-strain properties (elastic moduli, yield stress, etc). When this material definition is used with uncoupled stress/displacement elements, the heat transfer properties are ignored by Abaqus; when it is used with heat transfer elements, the mechanical strength properties are ignored. This capability allows you to develop complete material definitions and use them in any analysis.

Defining spatially varying material behavior for homogenous solid continuum elements using distributions in Abaqus/Standard

In Abaqus/Standard spatially varying mass density (“Density,” Section 21.2.1), linear elastic behavior (“Linear elastic behavior,” Section 22.2.1), and thermal expansion (“Thermal expansion,” Section 26.1.2) can be defined for homogeneous solid continuum elements using distributions (“Distribution definition,” Section 2.8.1). Using distributions in a model with significant variation in material behavior can greatly simplify pre- and postprocessing and improve performance during the analysis by allowing a single material definition to define the spatially varying material behavior. Without distributions such a model may require many material definitions and associated section assignments.

21.1.2 MATERIAL DATA DEFINITION

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Combining material behaviors," Section 21.1.3
- *MATERIAL
- "Creating materials," Section 12.4.1 of the Abaqus/CAE User's Manual

Overview

A material definition in Abaqus:

- specifies the behavior of a material and supplies all the relevant property data;
- can contain multiple material behaviors;
- is assigned a name, which is used to refer to those parts of the model that are made of that material;
- can have temperature and/or field variable dependence;
- can have solution variable dependence in Abaqus/Standard; and
- can be specified in a local coordinate system ("Orientations," Section 2.2.5), which is required if the material is not isotropic.

Material definitions

Any number of materials can be defined in an analysis. Each material definition can contain any number of material behaviors, as required, to specify the complete material behavior. For example, in a linear static stress analysis only elastic material behavior may be needed, while in a more complicated analysis several material behaviors may be required.

A name must be assigned to each material definition. This name allows the material to be referenced from the section definitions used to assign this material to regions in the model.

Input File Usage: *MATERIAL, NAME=*name*

Each material definition is specified in a data block, which is initiated by a *MATERIAL option. The material definition continues until an option that does not define a material behavior (such as another *MATERIAL option) is introduced, at which point the material definition is assumed to be complete. The order of the material behavior options is not important. All material behavior options within the data block are assumed to define the same material.

Abaqus/CAE Usage: Property module: material editor: **Name**
Use the menu bar under the **Material Options** list to add behaviors to a material.

Large-strain considerations

When giving material properties for finite-strain calculations, "stress" means "true" (Cauchy) stress (force per current area) and "strain" means logarithmic strain. For example, unless otherwise indicated, for uniaxial behavior

$$\varepsilon = \int \frac{dl}{l} = \ln\left(\frac{l}{l_0}\right).$$

Specifying material data as functions of temperature and independent field variables

Material data are often specified as functions of independent variables such as temperature. Material properties are made temperature dependent by specifying them at several different temperatures.

In some cases a material property can be defined as a function of variables calculated by Abaqus; for example, to define a work-hardening curve, stress must be given as a function of equivalent plastic strain.

Material properties can also be dependent on "field variables" (user-defined variables that can represent any independent quantity and are defined at the nodes, as functions of time). For example, material moduli can be functions of weave density in a composite or of phase fraction in an alloy. See "Specifying field variable dependence" for details. The initial values of field variables are given as initial conditions (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1) and can be modified as functions of time during an analysis (see "Predefined fields," Section 33.6.1). This capability is useful if, for example, material properties change with time because of irradiation or some other precalculated environmental effect.

Any material behaviors defined using a distribution in Abaqus/Standard (mass density, linear elastic behavior, and/or thermal expansion) cannot be defined with temperature and/or field dependence. However, material behaviors defined with distributions can be included in a material definition with other material behaviors that have temperature and/or field dependence. See "Density," Section 21.2.1; "Linear elastic behavior," Section 22.2.1; and "Thermal expansion," Section 26.1.2.

Interpolation of material data

In the simplest case of a constant property, only the constant value is entered. When the material data are functions of only one variable, the data must be given in order of increasing values of the independent variable. Abaqus then interpolates linearly for values between those given. The property is assumed to be constant outside the range of independent variables given (except for fabric materials, where it is extrapolated linearly outside the specified range using the slope at the last specified data point). Thus, you can give as many or as few input values as are necessary for the material model. If the material data depend on the independent variable in a strongly nonlinear manner, you must specify enough data points so that a linear interpolation captures the nonlinear behavior accurately.

When material properties depend on several variables, the variation of the properties with respect to the first variable must be given at fixed values of the other variables, in ascending values of the second variable, then of the third variable, and so on. The data must always be ordered so that the independent

variables are given increasing values. This process ensures that the value of the material property is completely and uniquely defined at any values of the independent variables upon which the property depends. See “Input syntax rules,” Section 1.2.1, for further explanation and an example.

Example: Temperature-dependent linear isotropic elasticity

Figure 21.1.2–1 shows a simple, isotropic, linear elastic material, giving the Young’s modulus and the Poisson’s ratio as functions of temperature.

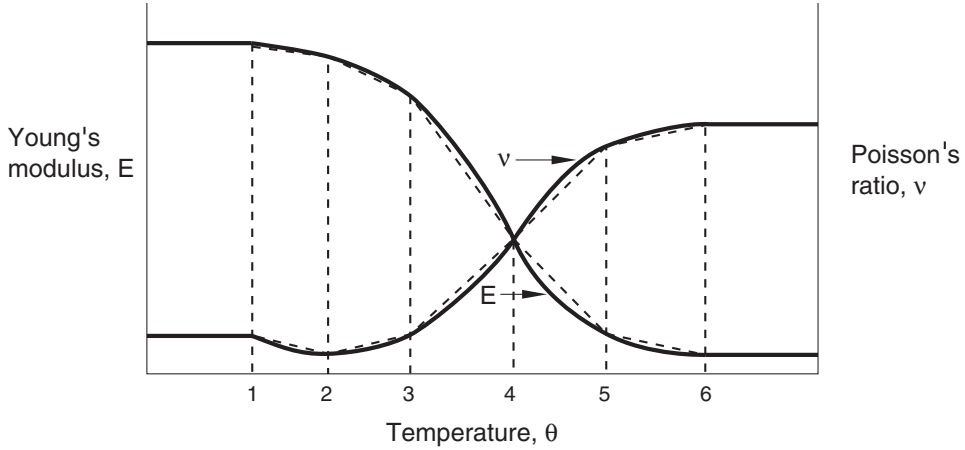


Figure 21.1.2–1 Example of material definition.

In this case six sets of values are used to specify the material description, as shown in the following table:

Elastic Modulus	Poisson’s Ratio	Temperature
E_1	ν_1	θ_1
E_2	ν_2	θ_2
E_3	ν_3	θ_3
E_4	ν_4	θ_4
E_5	ν_5	θ_5
E_6	ν_6	θ_6

For temperatures that are outside the range defined by θ_1 and θ_6, Abaqus assumes constant values for $\boldsymbol{E}$ and ν. The dotted lines on the graph represent the straight-line approximations that will be used for this model. In this example only one value of the thermal expansion coefficient is given, α_1, and it is independent of temperature.

Example: Elastic-plastic material

Figure 21.1.2–2 shows an elastic-plastic material for which the yield stress is dependent on the equivalent plastic strain and temperature.

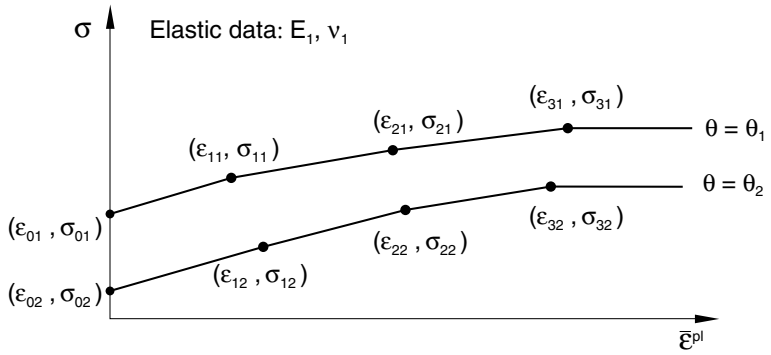


Figure 21.1.2–2 Example of material definition with two independent variables.

In this case the second independent variable (temperature) must be held constant, while the yield stress is described as a function of the first independent variable (equivalent plastic strain). Then, a higher value of temperature is chosen and the dependence on equivalent plastic strain is given at this temperature. This process, as shown in the following table, is repeated as often as necessary to describe the property variations in as much detail as required:

Yield Stress	Equivalent Plastic Strain	Temperature
σ_{01}	ε_{01}	θ_1
σ_{11}	ε_{11}	θ_1
σ_{21}	ε_{21}	θ_1
σ_{31}	ε_{31}	θ_1
σ_{02}	ε_{02}	θ_2
σ_{12}	ε_{12}	θ_2
σ_{22}	ε_{22}	θ_2
σ_{32}	ε_{32}	θ_2

Specifying field variable dependence

You can specify the number of user-defined field variable dependencies required for many material behaviors (see "Predefined fields," Section 33.6.1). If you do not specify a number of field variable

dependencies for a material behavior with which field variable dependence is available, the material data are assumed not to depend on field variables.

Input File Usage: **MATERIAL BEHAVIOR OPTION*, DEPENDENCIES=*n*

**MATERIAL BEHAVIOR OPTION* refers to any material behavior option for which field dependence can be specified. Each data line can hold up to eight data items. If more field variable dependencies are required than fit on a single data line, more data lines can be added. For example, a linear, isotropic elastic material can be defined as a function of temperature and seven field variables (fv) as follows:

```
*ELASTIC, TYPE=ISOTROPIC, DEPENDENCIES=7
E, ν, θ, fv1, fv2, fv3, fv4, fv5
fv6, fv7
```

This pair of data lines would be repeated as often as necessary to define the material as a function of the temperature and field variables.

Abaqus/CAE Usage: Property module: material editor: *material behavior*: **Number of field variables:** *n*

material behavior refers to any material behavior for which field dependence can be specified.

Specifying material data as functions of solution-dependent variables

In Abaqus you can introduce dependence on solution variables with a user subroutine. User subroutines **USDFLD** in Abaqus/Standard and **VUSDFLD** in Abaqus/Explicit allow you to define field variables at a material point as functions of time, of material directions, and of any of the available material point quantities: those listed in "Abaqus/Standard output variable identifiers," Section 4.2.1, for the case of **USDFLD**, and those listed in "Available output variable keys" in "Obtaining material point information in an Abaqus/Explicit analysis," Section 2.1.7 of the Abaqus User Subroutines Reference Manual, for the case of **VUSDFLD**. Material properties defined as functions of these field variables may, thus, be dependent on the solution.

User subroutines **USDFLD** and **VUSDFLD** are called at each material point for which the material definition includes a reference to the user subroutine.

For general analysis steps the values of variables provided in user subroutines **USDFLD** and **VUSDFLD** are those corresponding to the start of the increment. Hence, the solution dependence introduced in this way is explicit: the material properties for a given increment are not influenced by the results obtained during the increment. Consequently, the accuracy of the results will generally depend on the time increment size. This is usually not a concern in Abaqus/Explicit because the stable time increment is usually sufficiently small to ensure good accuracy. In Abaqus/Standard you can control the time increment from inside subroutine **USDFLD**. For linear perturbation steps the solution variables in the base state are available. (See "General and linear perturbation procedures," Section 6.1.3, for a discussion of general and linear perturbation steps.)

Input File Usage: *USER DEFINED FIELD

Abaqus/CAE Usage: User subroutines **USDFLD** and **VUSDFLD** are not supported in Abaqus/CAE.

Regularizing user-defined data in Abaqus/Explicit and Abaqus/CFD

Interpolating material data as functions of independent variables requires table lookups of the material data values during the analysis. The table lookups occur frequently in Abaqus/Explicit and Abaqus/CFD, and are most economical if the interpolation is from regular intervals of the independent variables. For example, the data shown in Figure 21.1.2–1 are not regular because the intervals in temperature (the independent variable) between adjacent data points vary. You are not required to specify regular material data. Abaqus/Explicit and Abaqus/CFD will automatically regularize user-defined data. For example, the temperature values in Figure 21.1.2–1 may be defined at 10°, 20°, 25°, 28°, 30°, and 35° C. In this case Abaqus/Explicit and Abaqus/CFD can regularize the data by defining the data over 25 increments of 1° C and your piecewise linear data will be reproduced exactly. This regularization requires the expansion of your data from values at 6 temperature points to values at 26 temperature points. This example is a case where a simple regularization can reproduce your data exactly.

If there are multiple independent variables, the concept of regular data also requires that the minimum and maximum values (the range) be constant for each independent variable while specifying the other independent variables. The material definition in Figure 21.1.2–2 illustrates a case where the material data are not regular since $\epsilon_{11} \neq \epsilon_{12}$, $\epsilon_{21} \neq \epsilon_{22}$, and $\epsilon_{31} \neq \epsilon_{32}$. Abaqus/Explicit will also regularize data involving multiple independent variables, although the data provided must satisfy the rules specified in "Input syntax rules," Section 1.2.1.

Error tolerance used in regularizing user-defined data

It is not always desirable to regularize the input data so that they are reproduced exactly in a piecewise linear manner. Suppose the yield stress is defined as a function of plastic strain in Abaqus/Explicit as follows:

Yield Stress	Plastic Strain
50000	.0
75000	.001
80000	.003
85000	.010
86000	1.0

It is possible to regularize the data exactly but it is not very economical, since it requires the subdivision of the data into 1000 regular intervals. Regularization is more difficult if the smallest interval you defined is small compared to the range of the independent variable.

Abaqus/Explicit and Abaqus/CFD use an error tolerance to regularize the input data. The number of intervals in the range of each independent variable is chosen such that the error between the piecewise linear regularized data and each of your defined points is less than the tolerance times the range of the dependent variable. In some cases the number of intervals becomes excessive and Abaqus/Explicit or

Abaqus/CFD cannot regularize the data using a reasonable number of intervals. The number of intervals considered reasonable depends on the number of intervals you define. If you defined 50 or less intervals, the maximum number of intervals used by Abaqus/Explicit and Abaqus/CFD for regularization is equal to 100 times the number of user-defined intervals. If you defined more than 50 intervals, the maximum number of intervals used for regularization is equal to 5000 plus 10 times the number of user-defined intervals above 50. If the number of intervals becomes excessive, the program stops during the data checking phase and issues an error message. You can either redefine the material data or change the tolerance value. The default tolerance is 0.03.

The yield stress data in the example above are a typical case where such an error message may be issued. In this case you can simply remove the last data point since it produces only a small difference in the ultimate yield value.

Input File Usage: *MATERIAL, RTOL=*tolerance*

Abaqus/CAE Usage: Property module: material editor: **General→Regularization**: **Rtol:** *tolerance*

Regularization of strain-rate-dependent data in Abaqus/Explicit

Since strain rate dependence of data is usually measured at logarithmic intervals, Abaqus/Explicit regularizes strain rate data using logarithmic intervals rather than uniformly spaced intervals by default. This will generally provide a better match to typical strain-rate-dependent curves. You can specify linear strain rate regularization to use uniform intervals for regularization of strain rate data. The use of linear strain rate regularization affects only the regularization of strain rate as an independent variable and is relevant only if one of the following behaviors is used to define the material data:

- low-density foams (“Low-density foams,” Section 22.9.1)
- rate-dependent metal plasticity (“Classical metal plasticity,” Section 23.2.1)
- rate-dependent viscoplasticity defined by yield stress ratios (“Rate-dependent yield,” Section 23.2.3)
- shear failure defined using direct tabular data (“Dynamic failure models,” Section 23.2.8)
- rate-dependent Drucker-Prager hardening (“Extended Drucker-Prager models,” Section 23.3.1)
- rate-dependent concrete damaged plasticity (“Concrete damaged plasticity,” Section 23.6.3)
- rate-dependent damage initiation criterion (“Damage initiation for ductile metals,” Section 24.2.2)

Input File Usage: Use the following option to specify logarithmic regularization (default):

*MATERIAL, STRAIN RATE REGULARIZATION=LOGARITHMIC

Use the following option to specify linear regularization:

*MATERIAL, STRAIN RATE REGULARIZATION=LINEAR

Abaqus/CAE Usage: Property module: material editor: **General→Regularization**: **Strain rate regularization**: **Logarithmic** or **Linear**

Evaluation of strain-rate-dependent data in Abaqus/Explicit

Rate-sensitive material constitutive behavior may introduce nonphysical high-frequency oscillations in an explicit dynamic analysis. To overcome this problem, Abaqus/Explicit computes the equivalent plastic strain rate used for the evaluation of strain-rate-dependent data as

$$\dot{\bar{\varepsilon}}^{pl}|_{t+\Delta t} = \omega \, \frac{\Delta \bar{\varepsilon}^{pl}}{\Delta t} + (1-\omega) \, \dot{\bar{\varepsilon}}^{pl}|_{t}.$$

Here $\Delta \bar{\varepsilon}^{pl}$ is the incremental change in equivalent plastic strain during the time increment Δt, and $\dot{\bar{\varepsilon}}^{pl}|_{t}$ and $\dot{\bar{\varepsilon}}^{pl}|_{t+\Delta t}$ are the strain rates at the beginning and end of the increment, respectively. The factor ω $(0 < \omega \leq 1)$ facilitates filtering high-frequency oscillations associated with strain-rate-dependent material behavior. You can specify the value of the strain rate factor, ω, directly. The default value is 0.9. A value of $\omega = 1$ does not provide the desired filtering effect and should be avoided.

Input File Usage: *MATERIAL, SRATE FACTOR=ω

Abaqus/CAE Usage: You cannot specify the value of the strain rate factor in Abaqus/CAE.

21.1.3 COMBINING MATERIAL BEHAVIORS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Material data definition," Section 21.1.2
- "Creating materials," Section 12.4.1 of the Abaqus/CAE User's Manual

Overview

Abaqus provides a broad range of possible material behaviors. A material is defined by choosing the appropriate behaviors for the purpose of an analysis. This section describes the general rules for combining material behaviors. Specific information for each material behavior is also summarized at the end of each material behavior description section in this chapter.

Some of the material behaviors in Abaqus are completely unrestricted: they can be used alone or together with other behaviors. For example, thermal properties such as conductivity can be used in any material definition. They will be used in an analysis if the material is associated with elements that can solve heat transfer problems and if the analysis procedure allows for the thermal equilibrium equation to be solved.

Some material behaviors in Abaqus require the presence of other material behaviors, and some exclude the use of other material behaviors. For example, metal plasticity requires the definition of elastic material behavior or an equation of state and excludes all other rate-independent plasticity behaviors.

Complete material definitions

Abaqus requires that the material be sufficiently defined to provide suitable properties for those elements with which the material is associated and for all of the analysis procedures through which the model will be run. Thus, a material associated with displacement or structural elements must include either a "Complete mechanical" category behavior or an "Elasticity" category behavior, as discussed below. In Abaqus/Explicit density ("Density," Section 21.2.1) is required for all materials except hydrostatic fluids.

It is not possible to modify or add to material definitions once an analysis is started. However, material definitions can be modified in an import analysis. For example, a static analysis can be run in Abaqus/Standard using a material definition that does not include a density specification. Density can be added to the material definition when the analysis is imported into Abaqus/Explicit.

All aspects of a material's behavior need not be fully defined; any behavior that is omitted is assumed not to exist in that part of the model. For example, if elastic material behavior is defined for a metal but metal plasticity is not defined, the material is assumed not to have a yield stress. You must ensure that the material is adequately defined for the purpose of the analysis. The material can include behaviors that are not relevant for the analysis, as described in "Material library: overview," Section 21.1.1. Thus,

you can include general material behavior libraries, without having to delete those behaviors that are not needed for a particular application. This generality offers great flexibility in material modeling.

In Abaqus/Standard any material behaviors defined using a distribution ("Distribution definition," Section 2.8.1) can be combined with almost all material behaviors in a manner identical to how they are combined when no distributions are used. For example, if the linear elastic material behavior is defined using a distribution, it can be combined with metal plasticity or any other material behavior that can normally be combined with linear elastic behavior. In addition, more than one material behavior defined with a distribution (linear elastic behavior and thermal expansion, for example) can be included in the same material definition. The only exception is that a material defined with concrete damaged plasticity ("Concrete damaged plasticity," Section 23.6.3) cannot have any material behaviors defined with a distribution.

Material behavior combination tables

The material behavior combination tables that follow explain which behaviors must be used together. The tables also show the material behaviors that cannot be combined. Behaviors designated with an (S) are available only in Abaqus/Standard; behaviors designated with an (E) are available only in Abaqus/Explicit.

The behaviors are assigned to categories because exclusions are best described in terms of those categories. Some of the categories require explanation:

- "Complete mechanical behaviors" are those behaviors in Abaqus that, individually, completely define a material's mechanical (stress-strain) behavior. A behavior in this category, therefore, excludes any other such behavior and also excludes any behavior that defines part of a material's mechanical behavior: those behaviors that belong to the elasticity and plasticity categories.
- "Elasticity, fabric, and equation of state behaviors" contains all of the basic elasticity behaviors in Abaqus. If a behavior from the "Complete mechanical behaviors" category is not used and mechanical behavior is required, a behavior must be selected from this category. This selection then excludes any other elasticity behavior.
- "Enhancements for elasticity behaviors" contains behaviors that extend the modeling provided by the elasticity behaviors in Abaqus.
- "Rate-independent plasticity behaviors" contains all of the basic plasticity behaviors in Abaqus except deformation plasticity, which is in the "Complete mechanical behaviors" category because it completely defines the material's mechanical behavior.
- "Rate-dependent plasticity behaviors" contains behaviors that extend the modeling provided by the rate-independent plasticity behaviors and by the linear elastic material behavior.

If elastic-plastic behavior must be modeled, you should select an appropriate plasticity behavior from one of the plasticity behaviors categories and an elasticity behavior from one of the elasticity behaviors categories.

General behaviors:

These behaviors are unrestricted.

Behavior	Keyword	Requires
Material damping	*DAMPING	Elasticity, fabric, hyperelasticity, hyperfoam, low-density foam, or anisotropic hyperelasticity (except when used with beam or shell general sections or substructures)
Density	*DENSITY	Required in Abaqus/Explicit, except for hydrostatic fluid elements
Solution-dependent state variables	*DEPVAR	
Thermal expansion	*EXPANSION	

Complete mechanical behaviors:

These behaviors are mutually exclusive and exclude all behaviors listed for elasticity, plasticity, and hydrostatic fluid behaviors, including all related enhancements.

Behavior	Keyword	Requires
Acoustic medium	*ACOUSTIC MEDIUM	Density
Deformation plasticity(S)	*DEFORMATION PLASTICITY	
Mechanical user material	*USER MATERIAL (, TYPE=MECHANICAL in Abaqus/Standard)	

Elasticity, fabric, and equation of state behaviors:

These behaviors are mutually exclusive.

Behavior	Keyword	Requires
Elasticity	*ELASTIC	
Equation of state(E)	*EOS	
Fabric(E)	*FABRIC	
Hyperelasticity	*HYPERELASTIC	
Hyperfoam	*HYPERFOAM	
Anisotropic hyperelasticity	*ANISOTROPIC HYPERELASTIC	
Hypoelasticity(S)	*HYPOELASTIC	

Behavior	Keyword	Requires
Porous elasticity (S)	*POROUS ELASTIC	
Low-density foam (E)	*LOW DENSITY FOAM	

Enhancements for elasticity behaviors:

Behavior	Keyword	Requires
Elastic shear behavior for an equation of state(E)	*ELASTIC, TYPE=SHEAR	Equation of state
Strain-based failure measures	*FAIL STRAIN	Elasticity
Stress-based failure measures	*FAIL STRESS	Elasticity
Hysteresis(S)	*HYSTERESIS	Hyperelasticity (excludes all plasticity behaviors and Mullins effect)
Mullins effect	*MULLINS EFFECT	Hyperelasticity (excludes hysteresis), hyperfoam or anisotropic hyperelasticity
Compressive failure theory(S)	*NO COMPRESSION	Elasticity
Tension failure theory(S)	*NO TENSION	Elasticity
Viscoelasticity	*VISCOELASTIC	Elasticity, hyperelasticity, or hyperfoam (excludes all plasticity behaviors and all associated plasticity enhancements); or anisotropic hyperelasticity
Shear viscosity for an equation of state(E)	*VISCOSITY	Equation of state

Rate-independent plasticity behaviors:

These behaviors are mutually exclusive.

Behavior	Keyword	Requires
Brittle cracking(E)	*BRITTLE CRACKING	Isotropic elasticity and brittle shear
Modified Drucker-Prager/Cap plasticity	*CAP PLASTICITY	Drucker-Prager/Cap plasticity hardening and isotropic elasticity or porous elasticity
Cast iron plasticity	*CAST IRON PLASTICITY	Cast iron compression hardening, cast iron tension hardening, and isotropic elasticity

Behavior	Keyword	Requires
Cam-clay plasticity	*CLAY PLASTICITY	Elasticity or porous elasticity (in Abaqus/Standard)
		Isotropic elasticity (in Abaqus/Explicit)
Concrete[S]	*CONCRETE	Isotropic elasticity
Concrete damaged plasticity	*CONCRETE DAMAGED PLASTICITY	Concrete compression hardening, concrete tension stiffening, and isotropic elasticity
Crushable foam plasticity	*CRUSHABLE FOAM	Crushable foam hardening and isotropic elasticity
Drucker-Prager plasticity	*DRUCKER PRAGER	Drucker-Prager hardening and isotropic elasticity or porous elasticity (in Abaqus/Standard)
		Drucker-Prager hardening and isotropic elasticity or the combination of an equation of state and isotropic linear elastic shear behavior for an equation of state (in Abaqus/Explicit)
Plastic compaction behavior for an equation of state[E]	*EOS COMPACTION	Linear $U_s - U_p$ equation of state
Jointed material[S]	*JOINTED MATERIAL	Isotropic elasticity and a local orientation
Mohr-Coulomb plasticity	*MOHR COULOMB	Mohr-Coulomb hardening and isotropic elasticity
Metal plasticity	*PLASTIC	Elasticity or hyperelasticity (in Abaqus/Standard)
		Isotropic elasticity, orthotropic elasticity (requires anisotropic yield), hyperelasticity, or the combination of an equation of state and isotropic linear elastic shear behavior for an equation of state (in Abaqus/Explicit)

Rate-dependent plasticity behaviors:

These behaviors are mutually exclusive, except metal creep and time-dependent volumetric swelling.

Behavior	Keyword	Requires
Cap creep(S)	*CAP CREEP	Elasticity, modified Drucker-Prager/Cap plasticity, and Drucker-Prager/Cap plasticity hardening
Metal creep(S)	*CREEP	Elasticity (except when used to define rate-dependent gasket behavior; excludes all rate-independent plasticity behaviors except metal plasticity)
Drucker-Prager creep(S)	*DRUCKER PRAGER CREEP	Elasticity, Drucker-Prager plasticity, and Drucker-Prager hardening
Metal plasticity	*PLASTIC, RATE	Elasticity or hyperelasticity (in Abaqus/Standard)
		Isotropic elasticity, orthotropic elasticity (requires anisotropic yield), hyperelasticity, or the combination of an equation of state and isotropic linear elastic shear behavior for an equation of state (in Abaqus/Explicit)
Nonlinear viscoelasticity(S)	*VISCOELASTIC, NONLINEAR	Hyperelasticity
Rate-dependent viscoplasticity	*RATE DEPENDENT	Drucker-Prager plasticity, crushable foam plasticity, or metal plasticity
Time-dependent volumetric swelling(S)	*SWELLING	Elasticity (excludes all rate-independent plasticity behaviors except metal plasticity)
Two-layer viscoplasticity(S)	*VISCOUS	Elasticity and metal plasticity

Enhancements for plasticity behaviors:

Behavior	Keyword	Requires
Annealing temperature	*ANNEAL TEMPERATURE	Metal plasticity
Brittle failure(E)	*BRITTLE FAILURE	Brittle cracking and brittle shear

Behavior	Keyword	Requires
Cyclic hardening	*CYCLIC HARDENING	Metal plasticity with nonlinear isotropic/kinematic hardening
Inelastic heat fraction	*INELASTIC HEAT FRACTION	Metal plasticity and specific heat
Oak Ridge National Laboratory constitutive model(S)	*ORNL	Metal plasticity, cycled yield stress data, and, usually, metal creep
Porous material failure criteria(E)	*POROUS FAILURE CRITERIA	Porous metal plasticity
Porous metal plasticity	*POROUS METAL PLASTICITY	Metal plasticity
Anisotropic yield/creep	*POTENTIAL	Metal plasticity, metal creep, or two-layer viscoplasticity
Shear failure(E)	*SHEAR FAILURE	Metal plasticity
Tension cutoff	*TENSION CUTOFF	Mohr-Coulomb plasticity

Enhancement for elasticity or plasticity behaviors:

Behavior	Keyword	Requires
Tensile failure(E)	*TENSILE FAILURE	Metal plasticity or equation of state
Damage initiation	*DAMAGE INITIATION	For elasticity behaviors: elasticity based on a traction-separation description for cohesive elements or elasticity model for fiber-reinforced composites For plasticity behaviors: elasticity and metal plasticity or Drucker-Prager plasticity
Damage evolution	*DAMAGE EVOLUTION	Damage initiation
Damage stabilization	*DAMAGE STABILIZATION	Damage evolution

Thermal behaviors:

These behaviors are unrestricted but exclude thermal user materials.

Behavior	Keyword	Requires
Thermal conductivity	*CONDUCTIVITY	
Volumetric heat generation[S]	*HEAT GENERATION	
Latent heat	*LATENT HEAT	Density
Specific heat	*SPECIFIC HEAT	Density

Complete thermal behavior:

This behavior is unrestricted but excludes the thermal behaviors in the previous table.

Behavior	Keyword	Requires
Thermal user material[S]	*USER MATERIAL, TYPE=THERMAL	Density

Pore fluid flow behaviors:

These behaviors are unrestricted.

Behavior	Keyword	Requires
Swelling gel[S]	*GEL	Permeability, porous bulk moduli, and absorption/exsorption behavior
Moisture-driven swelling[S]	*MOISTURE SWELLING	Permeability and absorption/exsorption behavior
Permeability[S]	*PERMEABILITY	
Porous bulk moduli[S]	*POROUS BULK MODULI	Permeability and either elasticity or porous elasticity
Absorption/exsorption behavior[S]	*SORPTION	Permeability

Electrical behaviors:

These behaviors are unrestricted.

Behavior	Keyword	Requires
Dielectricity[S]	*DIELECTRIC	
Electrical conductivity[S]	*ELECTRICAL CONDUCTIVITY	
Fraction of electric energy released as heat[S]	*JOULE HEAT FRACTION	
Piezoelectricity[S]	*PIEZOELECTRIC	

Mass diffusion behaviors:

These behaviors exclude all other behaviors.

Behavior	Keyword	Requires
Mass diffusivity[S]	*DIFFUSIVITY	Solubility
Solubility[S]	*SOLUBILITY	Mass diffusivity

Hydrostatic fluid behaviors:

Behavior	Keyword	Requires
Fluid bulk modulus[S]	*FLUID BULK MODULUS	Hydraulic fluid
Hydrostatic fluid density	*FLUID DENSITY	
Fluid thermal expansion coefficient[S]	*FLUID EXPANSION	Hydraulic fluid

21.2 General properties

- “Density,” Section 21.2.1

21.2.1 DENSITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- *DENSITY
- "Specifying material mass density," Section 12.8.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A material's mass density:

- must be defined in Abaqus/Standard for eigenfrequency and transient dynamic analysis, transient heat transfer analysis, adiabatic stress analysis, and acoustic analysis;
- must be defined in Abaqus/Standard for gravity, centrifugal, and rotary acceleration loading;
- must be defined in Abaqus/Explicit for all materials except hydrostatic fluids;
- must be defined in Abaqus/CFD for all fluids;
- can be specified as a function of temperature and predefined variables;
- can be distributed from nonstructural features (such as paint on sheet metal panels in a car) to the underlying elements using a nonstructural mass definition; and
- can be defined with a distribution for solid continuum elements in Abaqus/Standard.

Defining density

Density can be defined as a function of temperature and field variables. However, for all elements in Abaqus/Standard with the exception of acoustic, heat transfer, coupled temperature-displacement, and coupled thermal-electrical elements , the density is a function of the initial values of temperature and field variables and changes in volume only. It will not be updated if temperatures and field variables change during the analysis. For Abaqus/Explicit the exception includes acoustic elements only. For Abaqus/CFD the density is considered constant for incompressible flows.

For acoustic, heat transfer, coupled temperature-displacement, and coupled thermal-electrical elements in Abaqus/Standard and acoustic elements in Abaqus/Explicit, the density will be continually updated to the value corresponding to the current temperature and field variables.

In an Abaqus/Standard analysis a spatially varying mass density can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include a default value for the density. If a distribution is used, no dependencies on temperature and/or field variables for the density can be defined.

Input File Usage: Use either of the following options:

*DENSITY
*DENSITY, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **General→Density**

You can toggle on **Use temperature-dependent data** to define the density as a function of temperature and/or select the **Number of field variables** to define the density as a function of field variables.

Units

Since Abaqus has no built-in dimensions, you must ensure that the density is given in consistent units. The use of consistent units, and density in particular, is discussed in "Conventions," Section 1.2.2. If American or English units are used, you must be particularly careful that the density used is in units of ML^{-3}, where mass is defined in units of FT^2L^{-1}.

Elements

The density behavior described in this section is used to specify mass density for all elements, except rigid elements. Mass density for rigid elements is specified as part of the rigid body definition (see "Rigid elements," Section 30.3.1).

In Abaqus/Explicit a nonzero mass density must be defined for all elements that are not part of a rigid body.

In Abaqus/Standard density must be defined for heat transfer elements and acoustic elements; mass density can be defined for stress/displacement elements, coupled temperature-displacement elements, and elements including pore pressure. For elements that include pore pressure as a degree of freedom, the density of the dry material should be given for the porous medium in a coupled pore fluid flow/stress analysis.

If you have a complex density for an acoustic medium, you should enter its real part here and convert the imaginary part into a volumetric drag, as discussed in "Acoustic medium," Section 26.3.1.

The mass contribution from features that have negligible structural stiffness can be added to the model by smearing the mass over an element set that is typically adjacent to the nonstructural feature. The nonstructural mass can be specified in the form of a total mass value, a mass per unit volume, a mass per unit area, or a mass per unit length (see "Nonstructural mass definition," Section 2.7.1). A nonstructural mass definition contributes additional mass to the specified element set and does not alter the underlying material density.

22. Elastic Mechanical Properties

Overview	22.1
Linear elasticity	22.2
Porous elasticity	22.3
Hypoelasticity	22.4
Hyperelasticity	22.5
Stress softening in elastomers	22.6
Viscoelasticity	22.7
Nonlinear Viscoelasticity	22.8
Rate sensitive elastomeric foams	22.9

22.1 Overview

- “Elastic behavior: overview,” Section 22.1.1

22.1.1 ELASTIC BEHAVIOR: OVERVIEW

The material library in Abaqus includes several models of elastic behavior:

- **Linear elasticity:** Linear elasticity ("Linear elastic behavior," Section 22.2.1) is the simplest form of elasticity available in Abaqus. The linear elastic model can define isotropic, orthotropic, or anisotropic material behavior and is valid for small elastic strains.
- **Plane stress orthotropic failure:** Failure theories are provided ("Plane stress orthotropic failure measures," Section 22.2.3) for use with linear elasticity. They can be used to obtain postprocessed output requests.
- **Porous elasticity:** The porous elastic model in Abaqus/Standard ("Elastic behavior of porous materials," Section 22.3.1) is used for porous materials in which the volumetric part of the elastic strain varies with the logarithm of the equivalent pressure stress. This form of nonlinear elasticity is valid for small elastic strains.
- **Hypoelasticity:** The hypoelastic model in Abaqus/Standard ("Hypoelastic behavior," Section 22.4.1) is used for materials in which the rate of change of stress is defined by an elasticity matrix multiplying the rate of change of elastic strain, where the elasticity matrix is a function of the total elastic strain. This general, nonlinear elasticity is valid for small elastic strains.
- **Rubberlike hyperelasticity:** For rubberlike material at finite strain the hyperelastic model ("Hyperelastic behavior of rubberlike materials," Section 22.5.1) provides a general strain energy potential to describe the material behavior for nearly incompressible elastomers. This nonlinear elasticity model is valid for large elastic strains.
- **Foam hyperelasticity:** The hyperfoam model ("Hyperelastic behavior in elastomeric foams," Section 22.5.2) provides a general capability for elastomeric compressible foams at finite strains. This nonlinear elasticity model is valid for large strains (especially large volumetric changes). The low-density foam model in Abaqus/Explicit ("Low-density foams," Section 22.9.1) is a nonlinear viscoelastic model suitable for specifying strain-rate sensitive behavior of low-density elastomeric foams such as used in crash and impact applications. The foam plasticity model ("Crushable foam plasticity models," Section 23.3.5) should be used for foam materials that undergo permanent deformation.
- **Anisotropic hyperelasticity:** The anisotropic hyperelastic model ("Anisotropic hyperelastic behavior," Section 22.5.3) provides a general capability for modeling materials that exhibit highly anisotropic and nonlinear elastic behavior (such as biomedical soft tissues, fiber-reinforced elastomers, etc.). The model is valid for large elastic strains and captures the changes in the preferred material directions (or fiber directions) with deformation.
- **Fabric materials:** The fabric model in Abaqus/Explicit ("Fabric material behavior," Section 23.4.1) for woven fabrics captures the directional nature of the stiffness along the fill and the warp yarn directions. It also captures the shear response as the yarn directions rotate relative to each other. The model takes into account finite strains including large shear rotations. It captures the highly nonlinear elastic response of fabrics through the use of test data or a user subroutine, **`VFABRIC`** (see "VFABRIC," Section 1.2.3 of the Abaqus User Subroutines Reference Manual) for the material characterization. The test data based

fabric behavior can include nonlinear elasticity, permanent deformation, rate-dependent response, and damage accumulation.

- **Viscoelasticity:** The viscoelastic model is used to specify time-dependent material behavior ("Time domain viscoelasticity," Section 22.7.1). In Abaqus/Standard it is also used to specify frequency-dependent material behavior ("Frequency domain viscoelasticity," Section 22.7.2). It must be combined with linear elasticity, rubberlike hyperelasticity, or foam hyperelasticity.
- **Parallel network viscoelastic model:** The parallel network viscoelastic model in Abaqus/Standard ("Parallel network viscoelastic model," Section 22.8.2) is intended for modeling nonlinear viscous behavior for materials subjected to large strains, such as polymers. The model consists of multiple parallel elastic and viscoelastic networks. The elastic response is defined using the hyperelastic material model, and the viscous response is specified using the flow rule derived from a creep potential.
- **Hysteresis:** The hysteresis model in Abaqus/Standard ("Hysteresis in elastomers," Section 22.8.1) is used to specify rate-dependent behavior of elastomers. It is used in conjunction with hyperelasticity.
- **Mullins effect:** The Mullins effect model ("Mullins effect," Section 22.6.1) is used to specify stress softening of filled rubber elastomers due to damage, a phenomenon referred to as Mullins effect. The model can also be used to include permanent energy dissipation and stress softening effects in elastomeric foams ("Energy dissipation in elastomeric foams," Section 22.6.2). It is used in conjunction with rubberlike hyperelasticity or foam hyperelasticity.
- **No compression or no tension elasticity:** The no compression or no tension models in Abaqus/Standard ("No compression or no tension," Section 22.2.2) can be used when compressive or tensile principal stresses should not be generated. These options can be used only with linear elasticity.

Thermal strain

Thermal expansion can be introduced for any of the elasticity or fabric models ("Thermal expansion," Section 26.1.2).

Elastic strain magnitude

Except in the hyperelasticity and fabric material models, the stresses are always assumed to be small compared to the tangent modulus of the elasticity relationship; that is, the elastic strain must be small (less than 5%). The total strain can be arbitrarily large if inelastic response such as metal plasticity is included in the material definition.

For finite-strain calculations where the large strains are purely elastic, the fabric model (for woven fabrics), the hyperelastic model (for rubberlike behavior), or the foam hyperelasticity model (for elastomeric foams) should be used. The hyperelasticity and fabric models are the only models that give realistic predictions of actual material behavior at large elastic strains. The linear or, in Abaqus/Standard, porous elasticity models are appropriate in other cases where the large strains are inelastic.

In Abaqus/Standard the linear elastic, porous elastic, and hypoelastic models will exhibit poor convergence characteristics if the stresses reach levels of 50% or more of the elastic moduli; this

limitation is not serious in practical cases because these material models are not valid for the resulting large strains.

22.2 Linear elasticity

- “Linear elastic behavior,” Section 22.2.1
- “No compression or no tension,” Section 22.2.2
- “Plane stress orthotropic failure measures,” Section 22.2.3

22.2.1 LINEAR ELASTIC BEHAVIOR

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- *ELASTIC
- "Creating a linear elastic material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A linear elastic material model:

- is valid for small elastic strains (normally less than 5%);
- can be isotropic, orthotropic, or fully anisotropic;
- can have properties that depend on temperature and/or other field variables; and
- can be defined with a distribution for solid continuum elements in Abaqus/Standard.

Defining linear elastic material behavior

The total stress is defined from the total elastic strain as

$$\boldsymbol{\sigma} = \mathbf{D}^{el}\boldsymbol{\varepsilon}^{el},$$

where $\boldsymbol{\sigma}$ is the total stress ("true," or Cauchy stress in finite-strain problems), $\mathbf{D}^{el}$ is the fourth-order elasticity tensor, and $\boldsymbol{\varepsilon}^{el}$ is the total elastic strain (log strain in finite-strain problems). Do not use the linear elastic material definition when the elastic strains may become large; use a hyperelastic model instead. Even in finite-strain problems the elastic strains should still be small (less than 5%).

Defining linear elastic response for viscoelastic materials

The elastic response of a viscoelastic material ("Time domain viscoelasticity," Section 22.7.1) can be specified by defining either the instantaneous response or the long-term response of the material. To define the instantaneous response, experiments to determine the elastic constants have to be performed within time spans much shorter than the characteristic relaxation time of the material.

Input File Usage: *ELASTIC, MODULI=INSTANTANEOUS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Moduli time scale (for viscoelasticity)**: **Instantaneous**

If, on the other hand, the long-term elastic response is used, data from experiments have to be collected after time spans much longer than the characteristic relaxation time of the viscoelastic material. Long-term elastic response is the default elastic material behavior.

Input File Usage: *ELASTIC, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Moduli time scale (for viscoelasticity)**: **Long-term**

Directional dependence of linear elasticity

Depending on the number of symmetry planes for the elastic properties, a material can be classified as either *isotropic* (an infinite number of symmetry planes passing through every point) or *anisotropic* (no symmetry planes). Some materials have a restricted number of symmetry planes passing through every point; for example, *orthotropic* materials have two orthogonal symmetry planes for the elastic properties. The number of independent components of the elasticity tensor $\mathbf{D}^{el}$ depends on such symmetry properties. You define the level of anisotropy and method of defining the elastic properties, as described below. If the material is anisotropic, a local orientation ("Orientations," Section 2.2.5) must be used to define the direction of anisotropy.

Stability of a linear elastic material

Linear elastic materials must satisfy the conditions of material or Drucker stability (see the discussion on material stability in "Hyperelastic behavior of rubberlike materials," Section 22.5.1). Stability requires that the tensor $\mathbf{D}^{el}$ be positive definite, which leads to certain restrictions on the values of the elastic constants. The stress-strain relations for several different classes of material symmetries are given below. The appropriate restrictions on the elastic constants stemming from the stability criterion are also given.

Defining isotropic elasticity

The simplest form of linear elasticity is the isotropic case, and the stress-strain relationship is given by

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \begin{bmatrix} 1/E & -\nu/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}.$$

The elastic properties are completely defined by giving the Young's modulus, E, and the Poisson's ratio, ν. The shear modulus, G, can be expressed in terms of E and ν as $G = E/2(1+\nu)$. These parameters can be given as functions of temperature and of other predefined fields, if necessary.

In Abaqus/Standard spatially varying isotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for E and ν. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: *ELASTIC, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Isotropic**

Stability

The stability criterion requires that $E > 0$, $G > 0$, and $-1 < \nu < 0.5$. Values of Poisson's ratio approaching 0.5 result in nearly incompressible behavior. With the exception of plane stress cases (including membranes and shells) or beams and trusses, such values generally require the use of "hybrid" elements in Abaqus/Standard and generate high frequency noise and result in excessively small stable time increments in Abaqus/Explicit.

Defining orthotropic elasticity by specifying the engineering constants

Linear elasticity in an orthotropic material is most easily defined by giving the "engineering constants": the three moduli E_1, E_2, E_3; Poisson's ratios ν_{12}, ν_{13}, ν_{23}; and the shear moduli G_{12}, G_{13}, and G_{23} associated with the material's principal directions. These moduli define the elastic compliance according to

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{21}/E_2 & -\nu_{31}/E_3 & 0 & 0 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & -\nu_{32}/E_3 & 0 & 0 & 0 \\ -\nu_{13}/E_1 & -\nu_{23}/E_2 & 1/E_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{13} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{23} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}.$$

The quantity ν_{ij} has the physical interpretation of the Poisson's ratio that characterizes the transverse strain in the *j*-direction, when the material is stressed in the *i*-direction. In general, ν_{ij} is not equal to ν_{ji}: they are related by $\nu_{ij}/E_i = \nu_{ji}/E_j$. The engineering constants can also be given as functions of temperature and other predefined fields, if necessary.

In Abaqus/Standard spatially varying orthotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for the elastic moduli and Poisson's ratios. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: *ELASTIC, TYPE=ENGINEERING CONSTANTS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Engineering Constants**

Stability

Material stability requires

$$
\begin{aligned}
&E_1, E_2, E_3, G_{12}, G_{13}, G_{23} > 0,\\
&|\nu_{12}| < (E_1/E_2)^{1/2},\\
&|\nu_{13}| < (E_1/E_3)^{1/2},\\
&|\nu_{23}| < (E_2/E_3)^{1/2},\\
&1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13} > 0.
\end{aligned}
$$

When the left-hand side of the inequality approaches zero, the material exhibits incompressible behavior. Using the relations ν_{ij}/E_i=ν_{ji}/E_j, the second, third, and fourth restrictions in the above set can also be expressed as

$$
\begin{aligned}
&|\nu_{21}| < (E_2/E_1)^{1/2},\\
&|\nu_{31}| < (E_3/E_1)^{1/2},\\
&|\nu_{32}| < (E_3/E_2)^{1/2}.
\end{aligned}
$$

Defining transversely isotropic elasticity

A special subclass of orthotropy is *transverse isotropy*, which is characterized by a plane of isotropy at every point in the material. Assuming the 1–2 plane to be the plane of isotropy at every point, transverse isotropy requires that E_1=E_2=E_p, ν_{31}=ν_{32}=ν_{tp}, ν_{13}=ν_{23}=ν_{pt}, and G_{13}=G_{23}=G_t, where ***p*** and ***t*** stand for "in-plane" and "transverse," respectively. Thus, while ν_{tp} has the physical interpretation of the Poisson's ratio that characterizes the strain in the plane of isotropy resulting from stress normal to it, ν_{pt} characterizes the transverse strain in the direction normal to the plane of isotropy resulting from stress in the plane of isotropy. In general, the quantities ν_{tp} and ν_{pt} are not equal and are related by ν_{tp}/E_t=ν_{pt}/E_p. The stress-strain laws reduce to

$$
\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} =
\begin{bmatrix}
1/E_p & -\nu_p/E_p & -\nu_{tp}/E_t & 0 & 0 & 0 \\
-\nu_p/E_p & 1/E_p & -\nu_{tp}/E_t & 0 & 0 & 0 \\
-\nu_{pt}/E_p & -\nu_{pt}/E_p & 1/E_t & 0 & 0 & 0 \\
0 & 0 & 0 & 1/G_p & 0 & 0 \\
0 & 0 & 0 & 0 & 1/G_t & 0 \\
0 & 0 & 0 & 0 & 0 & 1/G_t
\end{bmatrix}
\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix},
$$

where G_p=$E_p/2(1+\nu_p)$ and the total number of independent constants is only five.

In Abaqus/Standard spatially varying transverse isotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for the elastic moduli and Poisson's ratio. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: *ELASTIC, TYPE=ENGINEERING CONSTANTS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Engineering Constants**

Stability

In the transversely isotropic case the stability relations for orthotropic elasticity simplify to

$$
\begin{aligned}
&E_p, E_t, G_p, G_t > 0,\\
&|\nu_p| < 1,\\
&|\nu_{pt}| < (E_p/E_t)^{1/2},\\
&|\nu_{tp}| < (E_t/E_p)^{1/2},\\
&1 - {\nu_p}^2 - 2\nu_{tp}\nu_{pt} - 2\nu_p\nu_{tp}\nu_{pt} > 0.
\end{aligned}
$$

Defining orthotropic elasticity in plane stress

Under plane stress conditions, such as in a shell element, only the values of E_1, E_2, ν_{12}, G_{12}, G_{13}, and G_{23} are required to define an orthotropic material. (In all of the plane stress elements in Abaqus the $(1, 2)$ surface is the surface of plane stress, so that the plane stress condition is $\sigma_{33} = 0$.) The shear moduli G_{13} and G_{23} are included because they may be required for modeling transverse shear deformation in a shell. The Poisson's ratio ν_{21} is implicitly given as $\nu_{21} = (E_2/E_1)\nu_{12}$. In this case the stress-strain relations for the in-plane components of the stress and strain are of the form

$$
\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} =
\begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix}
\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{Bmatrix}.
$$

In Abaqus/Standard spatially varying plane stress orthotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for the elastic moduli and Poisson's ratio. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: *ELASTIC, TYPE=LAMINA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Lamina**

Stability

Material stability for plane stress requires

$$
\begin{aligned}
&E_1, E_2, G_{12}, G_{13}, G_{23} > 0,\\
&|\nu_{12}| < (E_1/E_2)^{1/2}.
\end{aligned}
$$

Defining orthotropic elasticity by specifying the terms in the elastic stiffness matrix

Linear elasticity in an orthotropic material can also be defined by giving the nine independent elastic stiffness parameters, as functions of temperature and other predefined fields, if necessary. In this case the stress-strain relations are of the form

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} D_{1111} & D_{1122} & D_{1133} & 0 & 0 & 0 \\ & D_{2222} & D_{2233} & 0 & 0 & 0 \\ & & D_{3333} & 0 & 0 & 0 \\ & & & D_{1212} & 0 & 0 \\ & sym & & & D_{1313} & 0 \\ & & & & & D_{2323} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = [D^{el}] \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix}.$$

For an orthotropic material the engineering constants define the $\mathbf{D}$ matrix as

$$\begin{aligned} D_{1111} &= E_1(1-\nu_{23}\nu_{32})\Upsilon, \\ D_{2222} &= E_2(1-\nu_{13}\nu_{31})\Upsilon, \\ D_{3333} &= E_3(1-\nu_{12}\nu_{21})\Upsilon, \\ D_{1122} &= E_1(\nu_{21}+\nu_{31}\nu_{23})\Upsilon = E_2(\nu_{12}+\nu_{32}\nu_{13})\Upsilon, \\ D_{1133} &= E_1(\nu_{31}+\nu_{21}\nu_{32})\Upsilon = E_3(\nu_{13}+\nu_{12}\nu_{23})\Upsilon, \\ D_{2233} &= E_2(\nu_{32}+\nu_{12}\nu_{31})\Upsilon = E_3(\nu_{23}+\nu_{21}\nu_{13})\Upsilon, \\ D_{1212} &= G_{12}, \\ D_{1313} &= G_{13}, \\ D_{2323} &= G_{23}, \end{aligned}$$

where

$$\Upsilon = \frac{1}{1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{31}\nu_{13}-2\nu_{21}\nu_{32}\nu_{13}}.$$

When the material stiffness parameters (the D_{ijkl}) are given directly, Abaqus imposes the constraint $\sigma_{33} = 0$ for the plane stress case to reduce the material's stiffness matrix as required.

In Abaqus/Standard spatially varying orthotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for the elastic moduli. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: *ELASTIC, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Orthotropic**

Stability

The restrictions on the elastic constants due to material stability are

$$
\begin{aligned}
&D_{1111}, D_{2222}, D_{3333}, D_{1212}, D_{1313}, D_{2323} > 0,\\
&|D_{1122}| < (D_{1111}D_{2222})^{1/2},\\
&|D_{1133}| < (D_{1111}D_{3333})^{1/2},\\
&|D_{2233}| < (D_{2222}D_{3333})^{1/2},\\
&det(D^{el}) > 0.
\end{aligned}
$$

The last relation leads to

$$
D_{1111}D_{2222}D_{3333} + 2D_{1122}D_{1133}D_{2233} - D_{2222}{D_{1133}}^2 - D_{1111}{D_{2233}}^2 - D_{3333}{D_{1122}}^2 > 0.
$$

These restrictions in terms of the elastic stiffness parameters are equivalent to the restrictions in terms of the "engineering constants." Incompressible behavior results when the left-hand side of the inequality approaches zero.

Defining fully anisotropic elasticity

For fully anisotropic elasticity 21 independent elastic stiffness parameters are needed. The stress-strain relations are as follows:

$$
\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} =
\begin{bmatrix}
D_{1111} & D_{1122} & D_{1133} & D_{1112} & D_{1113} & D_{1123} \\
 & D_{2222} & D_{2233} & D_{2212} & D_{2213} & D_{2223} \\
 & & D_{3333} & D_{3312} & D_{3313} & D_{3323} \\
 & & & D_{1212} & D_{1213} & D_{1223} \\
 & sym & & & D_{1313} & D_{1323} \\
 & & & & & D_{2323}
\end{bmatrix}
\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = [D^{el}]
\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix}.
$$

When the material stiffness parameters (the D_{ijkl}) are given directly, Abaqus imposes the constraint $\sigma_{33} = 0$ for the plane stress case to reduce the material's stiffness matrix as required.

In Abaqus/Standard spatially varying anisotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for the elastic moduli. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: *ELASTIC, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Anisotropic**

Stability

The restrictions imposed upon the elastic constants by stability requirements are too complex to express in terms of simple equations. However, the requirement that $\mathbf{D}^{el}$ is positive definite requires that all of the eigenvalues of the elasticity matrix $[D^{el}]$ be positive.

Defining orthotropic elasticity for warping elements

For two-dimensional meshed models of solid cross-section Timoshenko beam elements modeled with warping elements (see "Meshed beam cross-sections," Section 10.6.1), Abaqus offers a linear elastic material definition that can have two different shear moduli in the user-specified material directions. In the user-specified directions the stress-strain relations are as follows:

$$\begin{Bmatrix} \sigma \\ \tau_1 \\ \tau_2 \end{Bmatrix} = \begin{bmatrix} E & & \\ & G_1 & \\ & & G_2 \end{bmatrix} \begin{Bmatrix} \varepsilon \\ \gamma_1 \\ \gamma_2 \end{Bmatrix}.$$

A local orientation is used to define the angle α between the global directions and the user-specified material directions. In the cross-section directions the stress-strain relations are as follows:

$$\begin{Bmatrix} \sigma \\ \tau_1 \\ \tau_2 \end{Bmatrix} = \begin{bmatrix} E & 0 & 0 \\ & G_1(\cos\alpha)^2 + G_2(\sin\alpha)^2 & (G_1 - G_2)\cos\alpha\sin\alpha \\ & sym & G_1(\sin\alpha)^2 + G_2(\cos\alpha)^2 \end{bmatrix} \begin{Bmatrix} \varepsilon \\ \gamma_1 \\ \gamma_2 \end{Bmatrix},$$

where σ represents the beam's axial stress and τ_1 and τ_2 represent two shear stresses.

Input File Usage: *ELASTIC, TYPE=TRACTION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Traction**

Stability

The stability criterion requires that $E > 0$, $G_1 > 0$, and $G_2 > 0$.

Defining elasticity in terms of tractions and separations for cohesive elements

For cohesive elements used to model bonded interfaces (see "Defining the constitutive response of cohesive elements using a traction-separation description," Section 32.5.6) Abaqus offers an elasticity definition that can be written directly in terms of the nominal tractions and the nominal strains. Both uncoupled and coupled behaviors are supported. For uncoupled behavior each traction component depends only on its conjugate nominal strain, while for coupled behavior the response is more general (as shown below). In the local element directions the stress-strain relations for uncoupled behavior are as follows:

$$\begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{bmatrix} K_{nn} & & \\ & K_{ss} & \\ & & K_{tt} \end{bmatrix} \begin{Bmatrix} \varepsilon_n \\ \varepsilon_s \\ \varepsilon_t \end{Bmatrix}.$$

The quantities t_n, t_s, and t_t represent the nominal tractions in the normal and the two local shear directions, respectively; while the quantities ε_n, ε_s, and ε_t represent the corresponding nominal strains. For coupled traction separation behavior the stress-strain relations are as follows:

$$\begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{bmatrix} K_{nn} & K_{ns} & K_{nt} \\ K_{ns} & K_{ss} & K_{st} \\ K_{nt} & K_{st} & K_{tt} \end{bmatrix} \begin{Bmatrix} \varepsilon_n \\ \varepsilon_s \\ \varepsilon_t \end{Bmatrix}.$$

Input File Usage: Use the following option to define uncoupled elastic behavior for cohesive elements:

*ELASTIC, TYPE=TRACTION

Use the following option to define coupled elastic behavior for cohesive elements:

*ELASTIC, TYPE=COUPLED TRACTION

Abaqus/CAE Usage: Use the following option to define uncoupled elastic behavior for cohesive elements:

Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Traction**

Use the following option to define coupled elastic behavior for cohesive elements:

Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Coupled Traction**

Stability

The stability criterion for uncoupled behavior requires that $K_{nn} > 0$, $K_{ss} > 0$, and $K_{tt} > 0$. For coupled behavior the stability criterion requires that:

$$K_{nn} > 0, \quad K_{ss} > 0, \quad K_{tt} > 0;$$

$$K_{ns} < \sqrt{K_{nn} K_{ss}};$$

$$K_{st} < \sqrt{K_{ss} K_{tt}};$$

$$K_{nt} < \sqrt{K_{nn} K_{tt}};$$

$$det \begin{bmatrix} K_{nn} & K_{ns} & K_{nt} \\ K_{ns} & K_{ss} & K_{st} \\ K_{nt} & K_{st} & K_{tt} \end{bmatrix} > 0.$$

Defining isotropic shear elasticity for equations of state in Abaqus/Explicit

Abaqus/Explicit allows you to define isotropic shear elasticity to describe the deviatoric response of materials whose volumetric response is governed by an equation of state ("Elastic shear behavior" in "Equation of state," Section 25.2.1). In this case the deviatoric stress-strain relationship is given by

$$\mathbf{S} = 2\mu\mathbf{e}^{el},$$

where $\mathbf{S}$ is the deviatoric stress and $\mathbf{e}^{el}$ is the deviatoric elastic strain. You must provide the elastic shear modulus, μ, when you define the elastic deviatoric behavior.

Input File Usage: *ELASTIC, TYPE=SHEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Type**: **Shear**

Elements

Linear elasticity can be used with any stress/displacement element or coupled temperature-displacement element in Abaqus. The exceptions are traction elasticity, which can be used only with warping elements and cohesive elements; coupled traction elasticity, which can be used only with cohesive elements; shear elasticity, which can be used only with solid (continuum) elements except plane stress elements; and, in Abaqus/Explicit, anisotropic elasticity, which is not supported for truss, rebar, pipe, and beam elements.

If the material is (almost) incompressible (Poisson's ratio $\nu > 0.49$ for isotropic elasticity), hybrid elements should be used in Abaqus/Standard. Compressible anisotropic elasticity should not be used with second-order hybrid continuum elements: inaccurate results and/or convergence problems may occur.

22.2.2 NO COMPRESSION OR NO TENSION

Products: Abaqus/Standard Abaqus/CAE

WARNING: Except when used with truss or beam elements, Abaqus/Standard does not form an exact material stiffness for this option. Therefore, the convergence can sometimes be slow.

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Linear elastic behavior," Section 22.2.1
- *NO COMPRESSION
- *NO TENSION
- "Specifying elastic material properties" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The no compression and no tension elasticity models:

- are used to modify the linear elasticity of the material so that compressive stress or tensile stress cannot be generated; and
- can be used only in conjunction with an elasticity definition.

Defining the modified elastic behavior

The modified elastic behavior is obtained by first solving for the principal stresses assuming linear elasticity and then setting the appropriate principal stress values to zero. The associated stiffness matrix components will also be set to zero. These models are not history dependent: the directions in which the principal stresses are set to zero are recalculated at every iteration.

The no compression effect for a one-dimensional stress case such as a truss or a layer of a beam in a plane is illustrated in Figure 22.2.2–1. No compression and no tension definitions modify only the elastic response of the material.

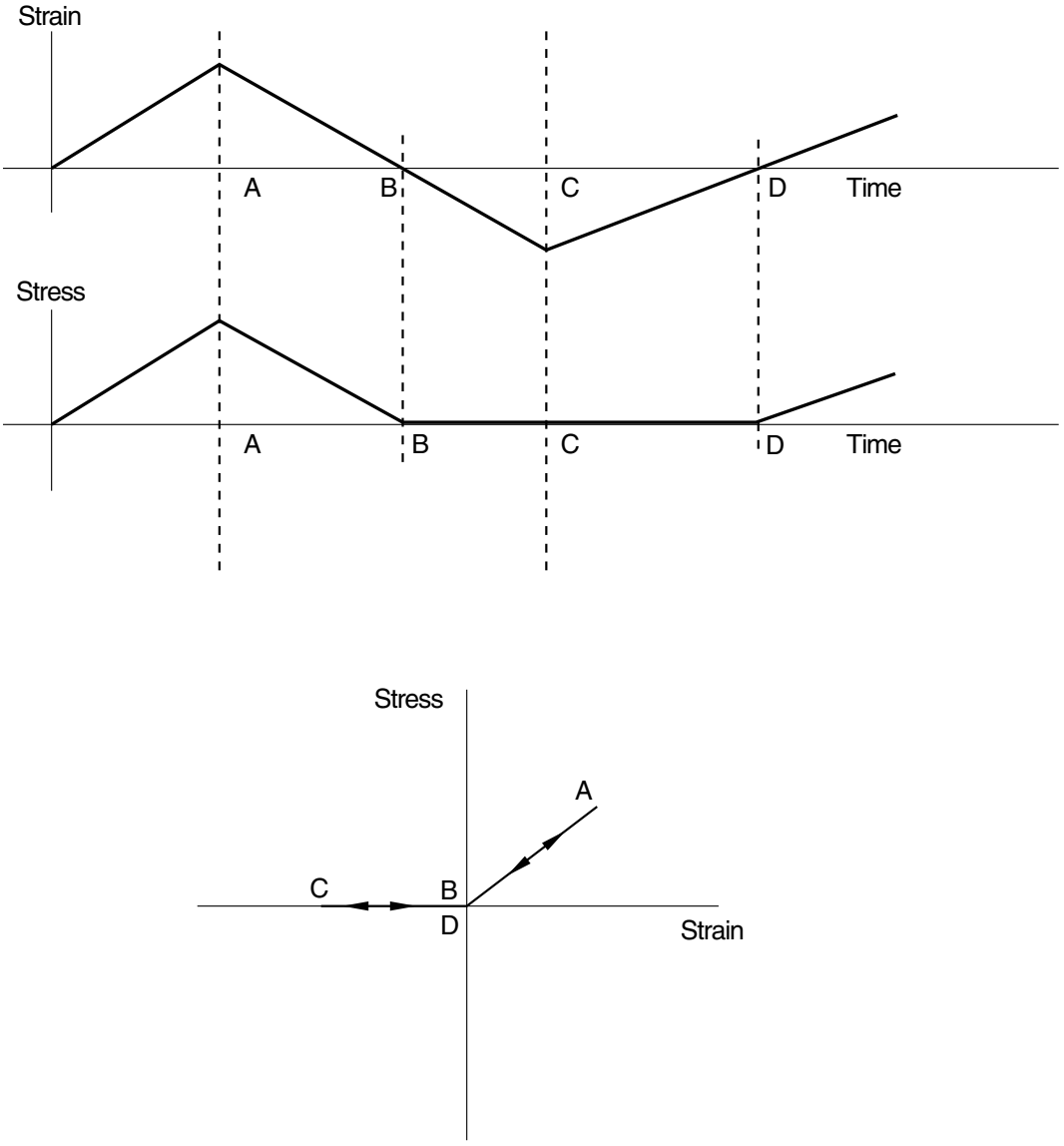


Figure 22.2.2–1 A no compression elastic case with an imposed strain cycle.

Input File Usage: Use one of the following options:

*NO COMPRESSION
*NO TENSION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **No compression** or **No tension**

Stability

Using no compression or no tension elasticity can make a model unstable: convergence difficulties may occur. Sometimes these difficulties can be overcome by overlaying each element that uses the no compression (or no tension) model with another element that uses a small value of Young's modulus (small in comparison with the Young's modulus of the element using modified elasticity). This technique creates a small "artificial" stiffness, which can stabilize the model.

Use with other material models

No compression and no tension definitions can be used only in conjunction with an elasticity definition. These definitions cannot be used with any other material option.

Elements

The no compression and no tension elasticity models can be used with any stress/displacement element in Abaqus/Standard. However, they cannot be used with shell elements or beam elements if section properties are pre-integrated using a general section definition.

22.2.3 PLANE STRESS ORTHOTROPIC FAILURE MEASURES

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Linear elastic behavior," Section 22.2.1
- *FAIL STRAIN
- *FAIL STRESS
- *ELASTIC
- "Defining stress-based failure measures for an elastic model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual
- "Defining strain-based failure measures for an elastic model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The orthotropic plane stress failure measures:

- are indications of material failure (normally used for fiber-reinforced composite materials; for alternative damage and failure models for fiber-reinforced composite materials, see "Damage and failure for fiber-reinforced composites: overview," Section 24.3.1);
- can be used only in conjunction with a linear elastic material model (with or without local material orientations);
- can be used for any element that uses a plane stress formulation; that is, for plane stress continuum elements, shell elements, and membrane elements;
- are postprocessed output requests and do not cause any material degradation; and
- take values that are greater than or equal to 0.0, with values that are greater than or equal to 1.0 implying failure.

Failure theories

Five different failure theories are provided: four stress-based theories and one strain-based theory.

We denote orthotropic material directions by 1 and 2, with the 1-material direction aligned with the fibers and the 2-material direction transverse to the fibers. For the failure theories to work correctly, the 1- and 2-directions of the user-defined elastic material constants must align with the fiber and the transverse-to-fiber directions, respectively. For applications other than fiber-reinforced composites, the 1- and 2-material directions should represent the strong and weak orthotropic-material directions, respectively.

In all cases tensile values must be positive and compressive values must be negative.

Stress-based failure theories

The input data for the stress-based failure theories are tensile and compressive stress limits, X_t and X_c, in the 1-direction; tensile and compressive stress limits, Y_t and Y_c, in the 2-direction; and shear strength (maximum shear stress), S, in the X–Y plane.

All four stress-based theories are defined and available with a single definition in Abaqus; the desired output is chosen by the output variables described at the end of this section.

Input File Usage: *FAIL STRESS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Suboptions→Fail Stress**

Maximum stress theory

If $\sigma_{11} > 0$, $X = X_t$; otherwise, $X = X_c$. If $\sigma_{22} > 0$, $Y = Y_t$; otherwise, $Y = Y_c$. The maximum stress failure criterion requires that

$$I_F = \max\left(\frac{\sigma_{11}}{X}, \frac{\sigma_{22}}{Y}, \left|\frac{\sigma_{12}}{S}\right|\right) < 1.0\,.$$

Tsai-Hill theory

If $\sigma_{11} > 0$, $X = X_t$; otherwise, $X = X_c$. If $\sigma_{22} > 0$, $Y = Y_t$; otherwise, $Y = Y_c$. The Tsai-Hill failure criterion requires that

$$I_F = \frac{\sigma_{11}^2}{X^2} - \frac{\sigma_{11}\sigma_{22}}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} < 1.0.$$

Tsai-Wu theory

The Tsai-Wu failure criterion requires that

$$I_F = F_1\sigma_{11} + F_2\sigma_{22} + F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{66}\sigma_{12}^2 + 2F_{12}\sigma_{11}\sigma_{22} < 1.0\,.$$

The Tsai-Wu coefficients are defined as follows:

$$F_1 = \frac{1}{X_t} + \frac{1}{X_c}\,, \quad F_2 = \frac{1}{Y_t} + \frac{1}{Y_c}\,, \quad F_{11} = -\frac{1}{X_t X_c}\,, \quad F_{22} = -\frac{1}{Y_t Y_c}\,, \quad F_{66} = \frac{1}{S^2}\,.$$

σ_{biax} is the equibiaxial stress at failure. If it is known, then

$$F_{12} = \frac{1}{2\sigma_{biax}^2}\left[1 - \left(\frac{1}{X_t} + \frac{1}{X_c} + \frac{1}{Y_t} + \frac{1}{Y_c}\right)\sigma_{biax} + \left(\frac{1}{X_t X_c} + \frac{1}{Y_t Y_c}\right)\sigma_{biax}^2\right];$$

otherwise,

$$F_{12} = \overset{*}{f}\sqrt{F_{11}F_{22}}\,,$$

where $-1.0 \leq \overset{*}{f} \leq 1.0$. The default value of $\overset{*}{f}$ is zero. For the Tsai-Wu failure criterion either $\overset{*}{f}$ or σ_{biax} must be given as input data. The coefficient $\overset{*}{f}$ is ignored if σ_{biax} is given.

Azzi-Tsai-Hill theory

The Azzi-Tsai-Hill failure theory is the same as the Tsai-Hill theory, except that the absolute value of the cross product term is taken:

$$I_F = \frac{\sigma_{11}^2}{X^2} - \frac{|\sigma_{11}\sigma_{22}|}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} < 1.0.$$

This difference between the two failure criteria shows up only when σ_{11} and σ_{22} have opposite signs.

Stress-based failure measures—failure envelopes

To illustrate the four stress-based failure measures, Figure 22.2.3–1, Figure 22.2.3–2, and Figure 22.2.3–3 show each failure envelope (i.e., $I_F = 1.0$) in (σ_{11}–σ_{22}) stress space compared to the Tsai-Hill envelope for a given value of in-plane shear stress. In each case the Tsai-Hill surface is the piecewise continuous elliptical surface with each quadrant of the surface defined by an ellipse centered at the origin. The parallelogram in Figure 22.2.3–1 defines the maximum stress surface. In Figure 22.2.3–2 the Tsai-Wu surface appears as the ellipse. In Figure 22.2.3–3 the Azzi-Tsai-Hill surface differs from the Tsai-Hill surface only in the second and fourth quadrants, where it is the outside bounding surface (i.e., further from the origin). Since all of the failure theories are calibrated by tensile and compressive failure under uniaxial stress, they all give the same values on the stress axes.

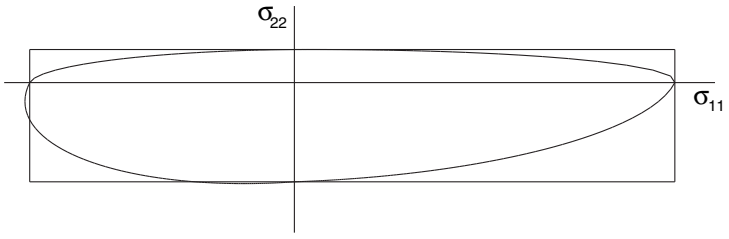


Figure 22.2.3–1 Tsai-Hill versus maximum stress failure envelope ($I_F = 1.0$).

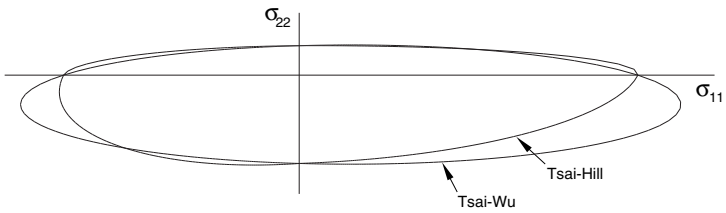


Figure 22.2.3–2 Tsai-Hill versus Tsai-Wu failure envelope ($I_F = 1.0$, $F_{12} = 0.0$).

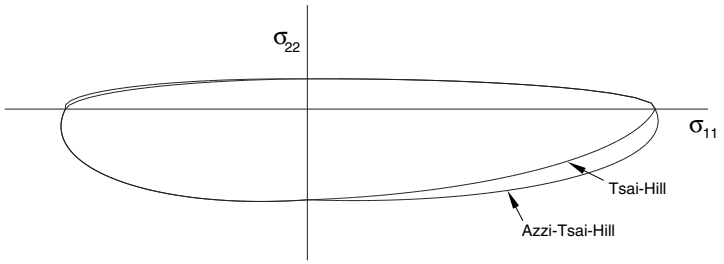


Figure 22.2.3–3 Tsai-Hill versus Azzi-Tsai-Hill failure envelope ($I_F = 1.0$).

Strain-based failure theory

The input data for the strain-based theory are tensile and compressive strain limits, X_{ε_t} and X_{ε_c}, in the 1-direction; tensile and compressive strain limits, Y_{ε_t} and Y_{ε_c}, in the 2-direction; and shear strain limit, S_ε, in the X–Y plane.

Input File Usage: *FAIL STRAIN

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**: **Suboptions→Fail Strain**

Maximum strain theory

If $\varepsilon_{11} > 0$, $X_\varepsilon = X_{\varepsilon_t}$; otherwise, $X_\varepsilon = X_{\varepsilon_c}$. If $\varepsilon_{22} > 0$, $Y_\varepsilon = Y_{\varepsilon_t}$; otherwise, $Y_\varepsilon = Y_{\varepsilon_c}$. The maximum strain failure criterion requires that

$$I_F = \max\left(\frac{\varepsilon_{11}}{X_\varepsilon}, \frac{\varepsilon_{22}}{Y_\varepsilon}, \left|\frac{\varepsilon_{12}}{S_\varepsilon}\right|\right) < 1.0\,.$$

Elements

The plane stress orthotropic failure measures can be used with any plane stress, shell, or membrane element in Abaqus.

Output

Abaqus provides output of the failure index, $\boldsymbol{R}$, if failure measures are defined with the material description. The definition of the failure index and the different output variables are described below.

Output failure indices

Each of the stress-based failure theories defines a failure surface surrounding the origin in the three-dimensional space $\{\sigma_{11}, \sigma_{22}, \sigma_{12}\}$. Failure occurs any time a state of stress is either on or outside this surface. The failure index, $\boldsymbol{R}$, is used to measure the proximity to the failure surface. $\boldsymbol{R}$ is defined as the scaling factor such that, for the given stress state $\{\sigma_{11}, \sigma_{22}, \sigma_{12}\}$,

$$\left\{\frac{\sigma_{11}}{R}, \frac{\sigma_{22}}{R}, \frac{\sigma_{12}}{R}\right\} \quad \Rightarrow \quad I_F = 1.0\,;$$

that is, $1/R$ is the scaling factor with which we need to multiply all of the stress components simultaneously to lie on the failure surface. Values $R < 1.0$ indicate that the state of stress is within the failure surface, while values $R \geq 1.0$ indicate failure. For the maximum stress theory $R \equiv I_F$.

The failure index $\boldsymbol{R}$ is defined similarly for the maximum strain failure theory. $\boldsymbol{R}$ is the scaling factor such that, for the given strain state $\{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}\}$,

$$\left\{\frac{\varepsilon_{11}}{R}, \frac{\varepsilon_{22}}{R}, \frac{\varepsilon_{12}}{R}\right\} \quad \Rightarrow \quad I_F = 1.0\,.$$

For the maximum strain theory $R \equiv I_F$.

Output variables

Output variable CFAILURE will provide output for all of the stress- and strain-based failure theories (see "Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2). In Abaqus/Standard history output can also be requested for the individual stress theories with output variables MSTRS, TSAIH, TSAIW, and AZZIT and for the strain theory with output variable MSTRN.

Output variables for the stress- and strain-based failure theories are always calculated at the material points of the element. In Abaqus/Standard element output can be requested at a location other than the material points (see "Output to the data and results files," Section 4.1.2); in this case the output variables are first calculated at the material points, then interpolated to the element centroid or extrapolated to the nodes.

22.3 Porous elasticity

- "Elastic behavior of porous materials," Section 22.3.1

22.3.1 ELASTIC BEHAVIOR OF POROUS MATERIALS

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- *POROUS ELASTIC
- *INITIAL CONDITIONS
- "Creating a porous elastic material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A porous elastic material model:

- is valid for small elastic strains (normally less than 5%);
- is a nonlinear, isotropic elasticity model in which the pressure stress varies as an exponential function of volumetric strain;
- allows a zero or nonzero elastic tensile stress limit; and
- can have properties that depend on temperature and other field variables.

Defining the volumetric behavior

Often, the elastic part of the volumetric behavior of porous materials is modeled accurately by assuming that the elastic part of the change in volume of the material is proportional to the logarithm of the pressure stress (Figure 22.3.1–1):

$$\frac{\kappa}{(1+e_0)}\ln\left(\frac{p_0+p_t^{el}}{p+p_t^{el}}\right)=J^{el}-1,$$

where κ is the "logarithmic bulk modulus"; e_0 is the initial void ratio; $\boldsymbol{p}$ is the equivalent pressure stress, defined by

$$p=-\frac{1}{3}\text{trace}\,\boldsymbol{\sigma}=-\frac{1}{3}(\sigma_{11}+\sigma_{22}+\sigma_{33});$$

p_0 is the initial value of the equivalent pressure stress; J^{el} is the elastic part of the volume ratio between the current and reference configurations; and p_t^{el} is the "elastic tensile strength" of the material (in the sense that $J^{el}\rightarrow\infty$ as $p\rightarrow -p_t^{el}$).

Input File Usage: Use all three of the following options to define a porous elastic material:

*POROUS ELASTIC, SHEAR=G or POISSON to define κ and p_t^{el}

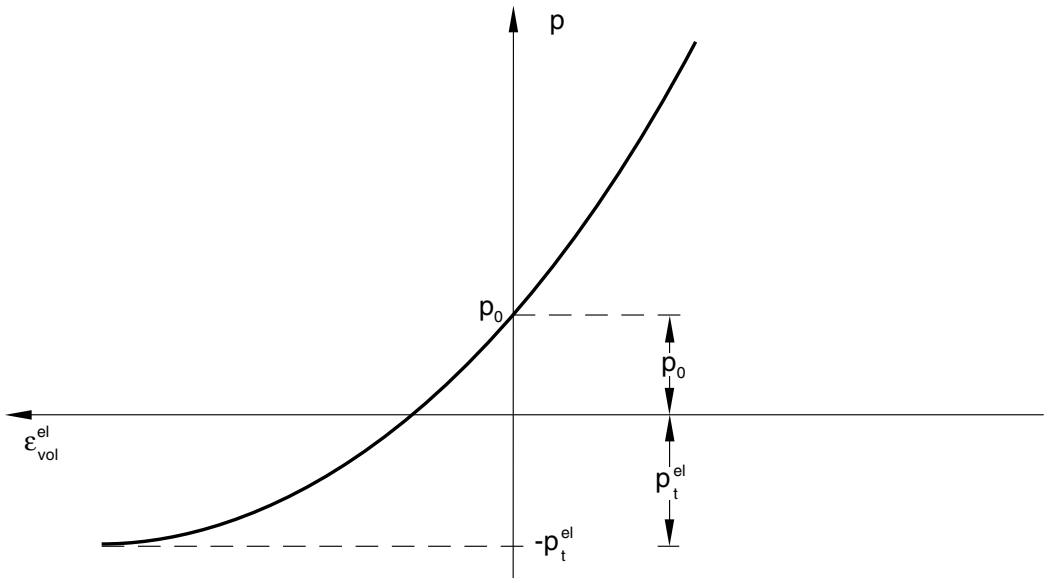


Figure 22.3.1–1 Porous elastic volumetric behavior.

∗INITIAL CONDITIONS, TYPE=STRESS to define p_0
∗INITIAL CONDITIONS, TYPE=RATIO to define e_0

Abaqus/CAE Usage: Use all three of the following options to define a porous elastic material:

Property module: material editor: **Mechanical→Elasticity→Porous Elastic**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Void ratio** for the **Types for Selected Step**

Defining the shear behavior

The deviatoric elastic behavior of a porous material can be defined in either of two ways.

By defining the shear modulus

Give the shear modulus, G. The deviatoric stress, $\mathbf{S}$, is then related to the deviatoric part of the total elastic strain, $\mathbf{e}^{el}$, by

$$\mathbf{S} = 2G\,\mathbf{e}^{el}.$$

In this case the shear behavior is not affected by compaction of the material.

Input File Usage: ∗POROUS ELASTIC, SHEAR=G

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Porous Elastic**: **Shear**: **G**

By defining Poisson's ratio

Define Poisson's ratio, ν. The instantaneous shear modulus is then defined from the instantaneous bulk modulus and Poisson's ratio as

$$G = \frac{3(1-2\nu)(1+e_0)}{2(1+\nu)\kappa}(p+p_t^{el})\exp(\varepsilon_{vol}^{el}),$$

where $\varepsilon_{vol}^{el} = \ln J^{el}$ is the logarithmic measure of the elastic volume change. In this case

$$d\mathbf{S} = 2G\, d\mathbf{e}^{el}.$$

Thus, the elastic shear stiffness increases as the material is compacted. This equation is integrated to give the total stress–total elastic strain relationship.

Input File Usage: *POROUS ELASTIC, SHEAR=POISSON

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Porous Elastic**: **Shear**: **Poisson**

Use with other material models

The porous elasticity model can be used by itself, or it can be combined with:

- the "Extended Drucker-Prager models," Section 23.3.1;
- the "Modified Drucker-Prager/Cap model," Section 23.3.2;
- the "Critical state (clay) plasticity model," Section 23.3.4; or
- isotropic expansion to introduce thermal volume changes ("Thermal expansion," Section 26.1.2).

It is not possible to use porous elasticity with rate-dependent plasticity or viscoelasticity.

Porous elasticity cannot be used with the porous metal plasticity model ("Porous metal plasticity," Section 23.2.9).

See "Combining material behaviors," Section 21.1.3, for more details.

Elements

Porous elasticity cannot be used with hybrid elements or plane stress elements (including shells and membranes), but it can be used with any other pure stress/displacement element in Abaqus/Standard.

If used with reduced-integration elements with total-stiffness hourglass control, Abaqus/Standard cannot calculate a default value for the hourglass stiffness of the element if the shear behavior is defined through Poisson's ratio. Hence, you must specify the hourglass stiffness. See "Section controls," Section 27.1.4, for details.

If fluid pore pressure is important (such as in undrained soils), stress/displacement elements that include pore pressure can be used.

22.4 Hypoelasticity

- “Hypoelastic behavior,” Section 22.4.1

22.4.1 HYPOELASTIC BEHAVIOR

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- *HYPOELASTIC
- "Creating a hypoelastic material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The hypoelastic material model:

- is valid for small elastic strains—the stresses should not be large compared to the elastic modulus of the material;
- is used when the load path is monotonic; and
- must be defined by user subroutine **UHYPEL** if temperature dependence is to be included.

Defining hypoelastic material behavior

In a hypoelastic material the rate of change of stress is defined as a tangent modulus matrix multiplying the rate of change of the elastic strain:

$$d\boldsymbol{\sigma} = \mathbf{D}^{el} : d\boldsymbol{\varepsilon}^{el},$$

where $d\boldsymbol{\sigma}$ is the rate of change of the stress (the "true," Cauchy, stress in finite-strain problems), $\mathbf{D}^{el}$ is the tangent elasticity matrix, and $d\boldsymbol{\varepsilon}^{el}$ is the rate of change of the elastic strain (the log strain in finite-strain problems).

Determining the hypoelastic material parameters

The entries in $\mathbf{D}^{el}$ are provided by giving Young's modulus, E, and Poisson's ratio, ν, as functions of strain invariants. The strain invariants are defined for this purpose as

$$\begin{aligned} I_1 &= \operatorname{trace} \boldsymbol{\varepsilon}^{el}, \\ I_2 &= \frac{1}{2}\left(\boldsymbol{\varepsilon}^{el} : \boldsymbol{\varepsilon}^{el} - I_1^2\right), \\ I_3 &= \det\left(\boldsymbol{\varepsilon}^{el}\right). \end{aligned}$$

You can define the material parameters directly or by using a user subroutine.

Direct specification

You can define the variation of Young's modulus and Poisson's ratio directly by specifying $\boldsymbol{E}$, ν, I_1, I_2, and I_3.

Input File Usage: *HYPOELASTIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hypoelastic**

User subroutine

If specifying $\boldsymbol{E}$ and ν as functions of the strain invariants directly does not allow sufficient flexibility, you can define the hypoelastic material by user subroutine **UHYPEL**.

Input File Usage: *HYPOELASTIC, USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hypoelastic**: **Use user subroutine UHYPEL**

Plane or uniaxial stress

For plane stress and uniaxial stress states Abaqus/Standard does not compute the out-of-plane strain components. For the purpose of defining the above invariants, it is assumed that $I_1 = 0$; that is, the material is assumed to be incompressible. For example, in a uniaxial stress case (such as a truss element) this assumption implies that

$$
\begin{aligned}
I_1 &= 0, \\
I_2 &= \frac{3}{4}\left(\varepsilon_{11}^{el}\right)^2, \\
I_3 &= \frac{1}{4}\left(\varepsilon_{11}^{el}\right)^3.
\end{aligned}
$$

Large-displacement analysis

For large-displacement analysis the strain measure in Abaqus is the integration of the rate of deformation. This strain measure corresponds to log strain if the principal directions do not rotate relative to the material. The strain invariant definitions should be interpreted in this way.

Use with other material models

The hypoelastic material model can be used only by itself in the material definition. It cannot be combined with viscoelasticity or with any inelastic response model. See "Combining material behaviors," Section 21.1.3, for more details.

Elements

The hypoelastic material model can be used with any of the stress/displacement elements in Abaqus/Standard.

22.5 Hyperelasticity

- “Hyperelastic behavior of rubberlike materials,” Section 22.5.1
- “Hyperelastic behavior in elastomeric foams,” Section 22.5.2
- “Anisotropic hyperelastic behavior,” Section 22.5.3

22.5.1 HYPERELASTIC BEHAVIOR OF RUBBERLIKE MATERIALS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Mullins effect," Section 22.6.1
- "Permanent set in rubberlike materials," Section 23.7.1
- *HYPERELASTIC
- *UNIAXIAL TEST DATA
- *BIAXIAL TEST DATA
- *PLANAR TEST DATA
- *VOLUMETRIC TEST DATA
- *MULLINS EFFECT
- "Creating an isotropic hyperelastic material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The hyperelastic material model:

- is isotropic and nonlinear;
- is valid for materials that exhibit instantaneous elastic response up to large strains (such as rubber, solid propellant, or other elastomeric materials); and
- requires that geometric nonlinearity be accounted for during the analysis step ("General and linear perturbation procedures," Section 6.1.3), since it is intended for finite-strain applications.

Compressibility

Most elastomers (solid, rubberlike materials) have very little compressibility compared to their shear flexibility. This behavior does not warrant special attention for plane stress, shell, membrane, beam, truss, or rebar elements, but the numerical solution can be quite sensitive to the degree of compressibility for three-dimensional solid, plane strain, and axisymmetric analysis elements. In cases where the material is highly confined (such as an O-ring used as a seal), the compressibility must be modeled correctly to obtain accurate results. In applications where the material is not highly confined, the degree of compressibility is typically not crucial; for example, it would be quite satisfactory in Abaqus/Standard to assume that the material is fully incompressible: the volume of the material cannot change except for thermal expansion.

Another class of rubberlike materials is elastomeric foam, which is elastic but very compressible. Elastomeric foams are discussed in "Hyperelastic behavior in elastomeric foams," Section 22.5.2.

We can assess the relative compressibility of a material by the ratio of its initial bulk modulus, K_0, to its initial shear modulus, μ_0. This ratio can also be expressed in terms of Poisson's ratio, ν, since

$$\nu = \frac{3K_0/\mu_0 - 2}{6K_0/\mu_0 + 2}.$$

The table below provides some representative values.

K_0/μ_0	**Poisson's ratio**
10	0.452
20	0.475
50	0.490
100	0.495
1000	0.4995
10,000	0.49995

Compressibility in Abaqus/Standard

In Abaqus/Standard the use of "hybrid" (mixed formulation) elements is recommended in both incompressible and almost incompressible cases. In plane stress, shell, and membrane elements the material is free to deform in the thickness direction. Similarly, in one-dimensional elements (such as beams, trusses, and rebars) the material is free to deform in the lateral directions. In these cases special treatment of the volumetric behavior is not necessary; the use of regular stress/displacement elements is satisfactory.

Compressibility in Abaqus/Explicit

Except for plane stress and uniaxial cases, it is not possible to assume that the material is fully incompressible in Abaqus/Explicit because the program has no mechanism for imposing such a constraint at each material calculation point. Instead, we must provide some compressibility. The difficulty is that, in many cases, the actual material behavior provides too little compressibility for the algorithms to work efficiently. Thus, except for plane stress and uniaxial cases, you must provide enough compressibility for the code to work, knowing that this makes the bulk behavior of the model softer than that of the actual material. Some judgment is, therefore, required to decide whether or not the solution is sufficiently accurate, or whether the problem can be modeled at all with Abaqus/Explicit because of this numerical limitation.

If no value is given for the material compressibility in the hyperelastic model, by default Abaqus/Explicit assumes $K_0/\mu_0 = 20$, corresponding to Poisson's ratio of 0.475. Since typical unfilled elastomers have K_0/μ_0 ratios in the range of 1,000 to 10,000 ($\nu = 0.4995$ to $\nu = 0.49995$) and filled elastomers have K_0/μ_0 ratios in the range of 50 to 200 ($\nu = 0.490$ to $\nu = 0.497$), this default provides much more compressibility than is available in most elastomers. However, if the elastomer is relatively unconfined, this softer modeling of the material's bulk behavior usually provides quite accurate results.

Unfortunately, in cases where the material is highly confined—such as when it is in contact with stiff, metal parts and has a very small amount of free surface, especially when the loading is highly compressive—it may not be feasible to obtain accurate results with Abaqus/Explicit.

If you are defining the compressibility rather than accepting the default value, an upper limit of 100 is suggested for the ratio of K_0/μ_0. Larger ratios introduce high frequency noise into the dynamic solution and require the use of excessively small time increments.

Isotropy assumption

In Abaqus all hyperelastic models are based on the assumption of isotropic behavior throughout the deformation history. Hence, the strain energy potential can be formulated as a function of the strain invariants.

Strain energy potentials

Hyperelastic materials are described in terms of a "strain energy potential," $U(\varepsilon)$, which defines the strain energy stored in the material per unit of reference volume (volume in the initial configuration) as a function of the strain at that point in the material. There are several forms of strain energy potentials available in Abaqus to model approximately incompressible isotropic elastomers: the Arruda-Boyce form, the Marlow form, the Mooney-Rivlin form, the neo-Hookean form, the Ogden form, the polynomial form, the reduced polynomial form, the Yeoh form, and the Van der Waals form. As will be pointed out below, the reduced polynomial and Mooney-Rivlin models can be viewed as particular cases of the polynomial model; the Yeoh and neo-Hookean potentials, in turn, can be viewed as special cases of the reduced polynomial model. Thus, we will occasionally refer collectively to these models as "polynomial models."

Generally, when data from multiple experimental tests are available (typically, this requires at least uniaxial and equibiaxial test data), the Ogden and Van der Waals forms are more accurate in fitting experimental results. If limited test data are available for calibration, the Arruda-Boyce, Van der Waals, Yeoh, or reduced polynomial forms provide reasonable behavior. When only one set of test data (uniaxial, equibiaxial, or planar test data) is available, the Marlow form is recommended. In this case a strain energy potential is constructed that will reproduce the test data exactly and that will have reasonable behavior in other deformation modes.

Evaluating hyperelastic materials

Abaqus/CAE allows you to evaluate hyperelastic material behavior by automatically creating response curves using selected strain energy potentials. In addition, you can provide experimental test data for a material without specifying a particular strain energy potential and have Abaqus/CAE evaluate the material to determine the optimal strain energy potential. See "Evaluating hyperelastic and viscoelastic material behavior," Section 12.4.7 of the Abaqus/CAE User's Manual, for details. Alternatively, you can use single-element test cases to evaluate the strain energy potential.

You can use single-element test cases to evaluate the strain energy potential.

Arruda-Boyce form

The form of the Arruda-Boyce strain energy potential is

$$U = \mu\left\{\frac{1}{2}(\overline{I}_1 - 3) + \frac{1}{20\lambda_m^2}(\overline{I}_1^2 - 9) + \frac{11}{1050\lambda_m^4}(\overline{I}_1^3 - 27) + \frac{19}{7000\lambda_m^6}(\overline{I}_1^4 - 81) + \frac{519}{673750\lambda_m^8}(\overline{I}_1^5 - 243)\right\} + \frac{1}{D}\left(\frac{J_{e\ell}^2 - 1}{2} - \ln J_{e\ell}\right),$$

where U is the strain energy per unit of reference volume; μ, λ_m, and D are temperature-dependent material parameters; $\overline{I}_1$ is the first deviatoric strain invariant defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2,$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus, μ_0, is related to μ with the expression

$$\mu_0 = \mu(1 + \frac{3}{5\lambda_m^2} + \frac{99}{175\lambda_m^4} + \frac{513}{875\lambda_m^6} + \frac{42039}{67375\lambda_m^8}).$$

A typical value of λ_m is 7, for which $\mu_0 = 1.0125\mu$. Both the initial shear modulus, μ_0, and the parameter μ are printed in the data (`.dat`) file if you request a printout of the model data from the analysis input file processor. The initial bulk modulus is related to D with the expression

$$K_0 = \frac{2}{D}.$$

Marlow form

The form of the Marlow strain energy potential is

$$U = U_{dev}(\overline{I}_1) + U_{vol}(J_{e\ell}),$$

where U is the strain energy per unit of reference volume, with U_{dev} as its deviatoric part and U_{vol} as its volumetric part; $\overline{I}_1$ is the first deviatoric strain invariant defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2,$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J_{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The deviatoric part of the potential is defined by providing either uniaxial, equibiaxial, or planar test data; while the volumetric part is defined by providing the volumetric test data, defining the Poisson's ratio, or specifying the lateral strains together with the uniaxial, equibiaxial, or planar test data.

Mooney-Rivlin form

The form of the Mooney-Rivlin strain energy potential is

$$U = C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3) + \frac{1}{D_1}(J^{e\ell} - 1)^2,$$

where U is the strain energy per unit of reference volume; C_{10}, C_{01}, and D_1 are temperature-dependent material parameters; $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2 \qquad \text{and} \qquad \overline{I}_2 = \overline{\lambda}_1^{(-2)} + \overline{\lambda}_2^{(-2)} + \overline{\lambda}_3^{(-2)},$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2(C_{10} + C_{01}), \qquad K_0 = \frac{2}{D_1}.$$

Neo-Hookean form

The form of the neo-Hookean strain energy potential is

$$U = C_{10}(\overline{I}_1 - 3) + \frac{1}{D_1}(J^{e\ell} - 1)^2,$$

where U is the strain energy per unit of reference volume; C_{10} and D_1 are temperature-dependent material parameters; $\overline{I}_1$ is the first deviatoric strain invariant defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2,$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2C_{10}, \qquad K_0 = \frac{2}{D_1}.$$

Ogden form

The form of the Ogden strain energy potential is

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}(\overline{\lambda}_1^{\alpha_i} + \overline{\lambda}_2^{\alpha_i} + \overline{\lambda}_3^{\alpha_i} - 3) + \sum_{i=1}^{N} \frac{1}{D_i}(J^{e\ell} - 1)^{2i},$$

where $\overline{\lambda}_i$ are the deviatoric principal stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; λ_i are the principal stretches; N is a material parameter; and μ_i, α_i, and D_i are temperature-dependent material parameters. The initial shear modulus and bulk modulus for the Ogden form are given by

$$\mu_0 = \sum_{i=1}^{N} \mu_i, \qquad K_0 = \frac{2}{D_1}.$$

The particular material models described above—the Mooney-Rivlin and neo-Hookean forms—can also be obtained from the general Ogden strain energy potential for special choices of μ_i and α_i.

Polynomial form

The form of the polynomial strain energy potential is

$$U = \sum_{i+j=1}^{N} C_{ij}(\overline{I}_1 - 3)^i(\overline{I}_2 - 3)^j + \sum_{i=1}^{N} \frac{1}{D_i}(J^{e\ell} - 1)^{2i},$$

where U is the strain energy per unit of reference volume; N is a material parameter; C_{ij} and D_i are temperature-dependent material parameters; $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2 \qquad \text{and} \qquad \overline{I}_2 = \overline{\lambda}_1^{(-2)} + \overline{\lambda}_2^{(-2)} + \overline{\lambda}_3^{(-2)},$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2(C_{10} + C_{01}), \qquad K_0 = \frac{2}{D_1}.$$

For cases where the nominal strains are small or only moderately large (< 100%), the first terms in the polynomial series usually provide a sufficiently accurate model. Some particular material models—the Mooney-Rivlin, neo-Hookean, and Yeoh forms—are obtained for special choices of C_{ij}.

Reduced polynomial form

The form of the reduced polynomial strain energy potential is

$$U = \sum_{i=1}^{N} C_{i0}(\overline{I}_1 - 3)^i + \sum_{i=1}^{N} \frac{1}{D_i}(J^{e\ell} - 1)^{2i},$$

where U is the strain energy per unit of reference volume; N is a material parameter; C_{i0} and D_i are temperature-dependent material parameters; $\overline{I}_1$ is the first deviatoric strain invariant defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2,$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2C_{10}, \qquad K_0 = \frac{2}{D_1}.$$

Van der Waals form

The form of the Van der Waals strain energy potential is

$$U = \mu\left\{-(\lambda_m^2 - 3)\left[\ln(1-\eta) + \eta\right] - \frac{2}{3}a\left(\frac{\tilde{I} - 3}{2}\right)^{\frac{3}{2}}\right\} + \frac{1}{D}\left(\frac{J_{e\ell}^2 - 1}{2} - \ln J_{e\ell}\right),$$

where

$$\tilde{I} = (1-\beta)\overline{I}_1 + \beta\overline{I}_2 \quad \text{and} \quad \eta = \sqrt{\frac{\tilde{I} - 3}{\lambda_m^2 - 3}}.$$

Here, U is the strain energy per unit of reference volume; μ is the initial shear modulus; λ_m is the locking stretch; a is the global interaction parameter; β is an invariant mixture parameter; and D governs the compressibility. These parameters can be temperature-dependent. $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2 \qquad \text{and} \qquad \overline{I}_2 = \overline{\lambda}_1^{(-2)} + \overline{\lambda}_2^{(-2)} + \overline{\lambda}_3^{(-2)},$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = \mu, \qquad K_0 = \frac{2}{D}.$$

Yeoh form

The form of the Yeoh strain energy potential is

$$\begin{aligned} U =& C_{10}(\overline{I}_1 - 3) + C_{20}(\overline{I}_1 - 3)^2 + C_{30}(\overline{I}_1 - 3)^3 \\ &+ \frac{1}{D_1}(J^{e\ell} - 1)^2 + \frac{1}{D_2}(J^{e\ell} - 1)^4 + \frac{1}{D_3}(J^{e\ell} - 1)^6, \end{aligned}$$

where U is the strain energy per unit of reference volume; C_{i0} and D_i are temperature-dependent material parameters; $\overline{I}_1$ is the first deviatoric strain invariant defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2,$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2C_{10}, \qquad K_0 = \frac{2}{D_1}.$$

Thermal expansion

Only isotropic thermal expansion is permitted with the hyperelastic material model.

The elastic volume ratio, $J^{e\ell}$, relates the total volume ratio, J, and the thermal volume ratio, J^{th}:

$$J^{e\ell} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = (1 + \varepsilon^{th})^3,$$

where ε^{th} is the linear thermal expansion strain that is obtained from the temperature and the isotropic thermal expansion coefficient ("Thermal expansion," Section 26.1.2).

Defining the hyperelastic material response

The mechanical response of a material is defined by choosing a strain energy potential to fit the particular material. The strain energy potential forms in Abaqus are written as separable functions of a deviatoric component and a volumetric component; i.e., $U = U_{dev}(\overline{I}_1, \overline{I}_2) + U_{vol}(J_{e\ell})$. Alternatively, in Abaqus/Standard you can define the strain energy potential with user subroutine **UHYPER**, in which case the strain energy potential need not be separable.

Generally for the hyperelastic material models available in Abaqus, you can either directly specify material coefficients or provide experimental test data and have Abaqus automatically determine appropriate values of the coefficients. An exception is the Marlow form: in this case the deviatoric part of the strain energy potential must be defined with test data. The different methods for defining the strain energy potential are described in detail below.

The properties of rubberlike materials can vary significantly from one batch to another; therefore, if data are used from several experiments, all of the experiments should be performed on specimens taken from the same batch of material, regardless of whether you or Abaqus compute the coefficients.

Viscoelastic and hysteretic materials

The elastic response of viscoelastic materials ("Time domain viscoelasticity," Section 22.7.1, and "Parallel network viscoelastic model," Section 22.8.2) and hysteretic materials ("Hysteresis in elastomers," Section 22.8.1) can be specified by defining either the instantaneous response or the long-term response of such materials. To define the instantaneous response, the experiments outlined in

the "Experimental tests" section that follows have to be performed within time spans much shorter than the characteristic relaxation times of these materials.

Input File Usage: *HYPERELASTIC, MODULI=INSTANTANEOUS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; any **Strain energy potential** except **Unknown**: **Moduli time scale (for viscoelasticity): Instantaneous**

If, on the other hand, the long-term elastic response is used, data from experiments have to be collected after time spans much longer than the characteristic relaxation times of these materials. Long-term elastic response is the default elastic material behavior.

Input File Usage: *HYPERELASTIC, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; any **Strain energy potential** except **Unknown**: **Moduli time scale (for viscoelasticity): Long-term**

Accounting for compressibility

Compressibility can be defined by specifying nonzero values for D_i (except for the Marlow model), by setting the Poisson's ratio to a value less than 0.5, or by providing test data that characterize the compressibility. The test data method is described later in this section. If you specify the Poisson's ratio for hyperelasticity other than the Marlow model, Abaqus computes the initial bulk modulus from the initial shear modulus

$$D_1 = \frac{2}{K_0} = \frac{3(1-2\nu)}{\mu_0(1+\nu)}.$$

For the Marlow model the specified Poisson's ratio represents a constant value, which determines the volumetric response throughout the deformation process. If D_1 is equal to zero, all of the D_i must be equal to zero. In such a case the material is assumed to be fully incompressible in Abaqus/Standard, while Abaqus/Explicit will assume compressible behavior with $K_0/\mu_0 = 20$ (Poisson's ratio of 0.475).

Input File Usage: *HYPERELASTIC, POISSON=ν

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; any **Strain energy potential** except **Unknown** or **User-defined**: **Input source: Test data**: **Poisson's ratio:** ν

Specifying material coefficients directly

The parameters of the hyperelastic strain energy potentials can be given directly as functions of temperature for all forms of the strain energy potential except the Marlow form.

Input File Usage: Use one of the following options:

*HYPERELASTIC, ARRUDA-BOYCE
*HYPERELASTIC, MOONEY-RIVLIN
*HYPERELASTIC, NEO HOOKE

∗HYPERELASTIC, OGDEN, N=*n* ($n \leq 6$)
∗HYPERELASTIC, POLYNOMIAL, N=*n* ($n \leq 6$)
∗HYPERELASTIC, REDUCED POLYNOMIAL, N=*n* ($n \leq 6$)
∗HYPERELASTIC, VAN DER WAALS
∗HYPERELASTIC, YEOH

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Coefficients** and **Strain energy potential: Arruda-Boyce**, **Mooney-Rivlin**, **Neo Hooke**, **Ogden**, **Polynomial**, **Reduced Polynomial**, **Van der Waals**, or **Yeoh**

Using test data to calibrate material coefficients

The material coefficients of the hyperelastic models can be calibrated by Abaqus from experimental stress-strain data. In the case of the Marlow model, the test data directly characterize the strain energy potential (there are no material coefficients for this model); the Marlow model is described in detail below. The value of ***N*** and experimental stress-strain data can be specified for up to four simple tests: uniaxial, equibiaxial, planar, and, if the material is compressible, a volumetric compression test. Abaqus will then compute the material parameters. The material constants are determined through a least-squares-fit procedure, which minimizes the relative error in stress. For the ***n*** nominal-stress–nominal-strain data pairs, the relative error measure ***E*** is minimized, where

$$E = \sum_{i=1}^{n} \left(1 - T_i^{\text{th}}/T_i^{\text{test}}\right)^2.$$

T_i^{test} is a stress value from the test data, and T_i^{th} comes from one of the nominal stress expressions derived below (see "Experimental tests"). Abaqus minimizes the relative error rather than an absolute error measure since this provides a better fit at lower strains. This method is available for all strain energy potentials and any order of ***N*** except for the polynomial form, where a maximum of $N = 2$ is allowed. The polynomial models are linear in terms of the constants C_{ij}; therefore, a linear least-squares procedure can be used. The Arruda-Boyce, Ogden, and Van der Waals potentials are nonlinear in some of their coefficients, thus necessitating the use of a nonlinear least-squares procedure. "Fitting of hyperelastic and hyperfoam constants," Section 4.6.2 of the Abaqus Theory Manual, contains a detailed derivation of the related equations.

It is generally best to obtain data from several experiments involving different kinds of deformation over the range of strains of interest in the actual application and to use all of these data to determine the parameters. This is particularly true for the phenomenological models; i.e., the Ogden and the polynomial models. It has been observed that to achieve good accuracy and stability, it is necessary to fit these models using test data from more than one deformation state. In some cases, especially at large strains, removing the dependence on the second invariant may alleviate this limitation. The Arruda-Boyce, neo-Hookean, and Van der Waals models with $\beta = 0$ offer a physical interpretation and provide a better prediction of general deformation modes when the parameters are based on only one test. An extensive discussion of this topic can be found in "Hyperelastic material behavior," Section 4.6.1 of the Abaqus Theory Manual.

This method does not allow the hyperelastic properties to be temperature dependent. However, if temperature-dependent test data are available, several curve fits can be conducted by performing a data check analysis on a simple input file. The temperature-dependent coefficients determined by Abaqus can then be entered directly in the actual analysis run.

Optionally, the parameter β in the Van der Waals model can be set to a fixed value while the other parameters are found using a least-squares curve fit.

As many data points as required can be entered from each test. It is recommended that data from all four tests (on samples taken from the same piece of material) be included and that the data points cover the range of nominal strains expected to arise in the actual loading. For the (general) polynomial and Ogden models and for the coefficient β in the Van der Waals model, the planar test data must be accompanied by the uniaxial test data, the biaxial test data, or both of these types of test data; otherwise, the solution to the least-squares fit will not be unique.

The strain data should be given as nominal strain values (change in length per unit of original length). For the uniaxial, equibiaxial, and planar tests stress data are given as nominal stress values (force per unit of original cross-sectional area). These tests allow for entering both compression and tension data. Compressive stresses and strains are entered as negative values.

If compressibility is to be specified, the D_i or $\boldsymbol{D}$ can be computed from volumetric compression test data. Alternatively, compressibility can be defined by specifying a Poisson's ratio, in which case Abaqus computes the bulk modulus from the initial shear modulus. If no such data are given, Abaqus/Standard assumes that $\boldsymbol{D}$ or all of the D_i are zero, whereas Abaqus/Explicit assumes compressibility corresponding to a Poisson's ratio of 0.475 (see "Compressibility in Abaqus/Explicit" above). For these compression tests the stress data are given as pressure values.

Input File Usage: Use one of the following options to select the strain energy potential:

*HYPERELASTIC, TEST DATA INPUT, ARRUDA-BOYCE
*HYPERELASTIC, TEST DATA INPUT, MOONEY-RIVLIN
*HYPERELASTIC, TEST DATA INPUT, NEO HOOKE
*HYPERELASTIC, TEST DATA INPUT, OGDEN, N=*n* ($n \leq 6$)
*HYPERELASTIC, TEST DATA INPUT, POLYNOMIAL, N=*n* ($n \leq 2$)
*HYPERELASTIC, TEST DATA INPUT, REDUCED POLYNOMIAL, N=*n* ($n \leq 6$)
*HYPERELASTIC, TEST DATA INPUT, VAN DER WAALS
*HYPERELASTIC, TEST DATA INPUT, VAN DER WAALS, BETA=β ($0 \leq \beta \leq 1$)
*HYPERELASTIC, TEST DATA INPUT, YEOH

In addition, use at least one and up to four of the following options to give the test data (see "Experimental tests" below):

*UNIAXIAL TEST DATA
*BIAXIAL TEST DATA
*PLANAR TEST DATA
*VOLUMETRIC TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**:

Material type: Isotropic; **Input source: Test data** and **Strain energy potential: Arruda-Boyce**, **Mooney-Rivlin**, **Neo Hooke**, **Ogden**, **Polynomial**, **Reduced Polynomial**, **Van der Waals** (**Beta: Fitted value** or **Specify**), or **Yeoh**

In addition, use at least one and up to four of the following options to give the test data (see "Experimental tests" below):

Test Data→Uniaxial Test Data
Test Data→Biaxial Test Data
Test Data→Planar Test Data
Test Data→Volumetric Test Data

Alternatively, you can select **Strain energy potential: Unknown** to define the material temporarily without specifying a particular strain energy potential. Then select **Material→Evaluate** to have Abaqus/CAE evaluate the material to determine the optimal strain energy potential.

Specifying the Marlow model

The Marlow model assumes that the strain energy potential is independent of the second deviatoric invariant $\overline{I}_2$. This model is defined by providing test data that define the deviatoric behavior, and, optionally, the volumetric behavior if compressibility must be taken into account. Abaqus will construct a strain energy potential that reproduces the test data exactly, as shown in Figure 22.5.1–1.

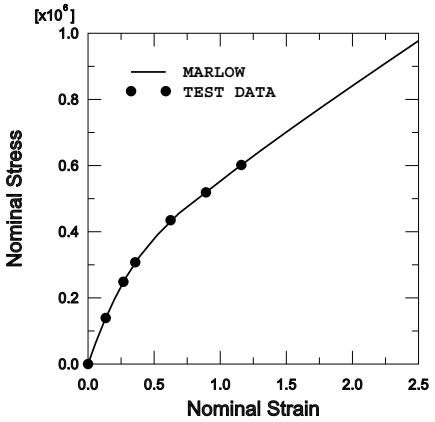


Figure 22.5.1–1 The results of the Marlow model with test data.

The interpolation and extrapolation of stress-strain data with the Marlow model is approximately linear for small and large strains. For intermediate strains in the range 0.1 to 1.0 a noticeable degree of nonlinearity may be observed in the interpolation/extrapolation with the Marlow model; for example, some nonlinearity is apparent between the 4th and 5th data points in Figure 22.5.1–1. To minimize undesirable nonlinearity, make sure that enough data points are specified in the intermediate strain range.

The deviatoric behavior is defined by specifying uniaxial, biaxial, or planar test data. Generally, you can specify either the data from tension tests or the data from compression tests because the tests are equivalent (see "Equivalent experimental tests). However, for beams, trusses, and rebars, the data from tension and compression tests can be specified together. Volumetric behavior is defined by using one of the following three methods:

- Specify nominal lateral strains, in addition to nominal stresses and nominal strains, as part of the uniaxial, biaxial, or planar test data.
- Specify Poisson's ratio for the hyperelastic material.
- Specify volumetric test data directly. Both hydrostatic tension and hydrostatic compression data can be specified. If only hydrostatic compression data are available, as is usually the case, Abaqus will assume that the hydrostatic pressure is an antisymmetric function of the nominal volumetric strain, $\epsilon_{vol} = J_{vol} - 1$.

If you do not define volumetric behavior, Abaqus/Standard assumes fully incompressible behavior, while Abaqus/Explicit assumes compressibility corresponding to a Poisson's ratio of 0.475.

Material test data in which the stress does not vary smoothly with increasing strain may lead to convergence difficulty during the simulation. It is highly recommended that smooth test data be used to define the Marlow form. Abaqus provides a smoothing algorithm, which is described in detail later in this section.

The test data for the Marlow model can also be given as a function of temperature and field variables. You must specify the number of user-defined field variable dependencies required.

Uniaxial, biaxial, and planar test data must be given in ascending order of the nominal strains; volumetric test data must be given in descending order of the volume ratio.

Input File Usage: To define the Marlow test data as a function of temperature and/or field variables, use the following option:

*HYPERELASTIC, MARLOW

with one of the following first three options and, optionally, the fourth option:

*UNIAXIAL TEST DATA, DEPENDENCIES=*n*
*BIAXIAL TEST DATA, DEPENDENCIES=*n*
*PLANAR TEST DATA, DEPENDENCIES=*n*
*VOLUMETRIC TEST DATA, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Strain energy potential: Marlow**

In addition, select one of the following first three options and, optionally, the fourth option to give the test data (see "Experimental tests" below):

Test Data→Uniaxial Test Data
Test Data→Biaxial Test Data
Test Data→Planar Test Data
Test Data→Volumetric Test Data

In each of the **Test Data Editor** dialog boxes, you can toggle on **Use temperature-dependent data** to define the test data as a function of temperature and/or select the **Number of field variables** to define the test data as a function of field variables.

Alternatively, you can select **Material→Evaluate** to have Abaqus/CAE evaluate the material. If you included temperature dependencies, field variable dependencies, or lateral nominal strain in the test data—which can only be defined in the **Marlow** hyperelastic definition—**Marlow** will be the only strain energy potential available for evaluation.

User subroutine specification in Abaqus/Standard

An alternative method provided in Abaqus/Standard for defining the hyperelastic material parameters allows the strain energy potential to be defined in user subroutine **UHYPER**. Either compressible or incompressible behavior can be specified. Optionally, you can specify the number of property values needed as data in the user subroutine. The derivatives of the strain energy potential with respect to the strain invariants must be provided directly through user subroutine **UHYPER**. If needed, you can specify the number of solution-dependent variables (see "User subroutines: overview," Section 18.1.1).

Input File Usage: Use one of the following two options:

*HYPERELASTIC, USER, TYPE=COMPRESSIBLE, PROPERTIES=*n*
*HYPERELASTIC, USER, TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Coefficients** and **Strain energy potential: User-defined**: optionally, toggle on **Include compressibility** and/or specify the **Number of property values**

Experimental tests

For a homogeneous material, homogeneous deformation modes suffice to characterize the material constants. Abaqus accepts test data from the following deformation modes:

- Uniaxial tension and compression
- Equibiaxial tension and compression
- Planar tension and compression (also known as pure shear)
- Volumetric tension and compression

These modes are illustrated schematically in Figure 22.5.1–2 and are described below. The most commonly performed experiments are uniaxial tension, uniaxial compression, and planar tension.

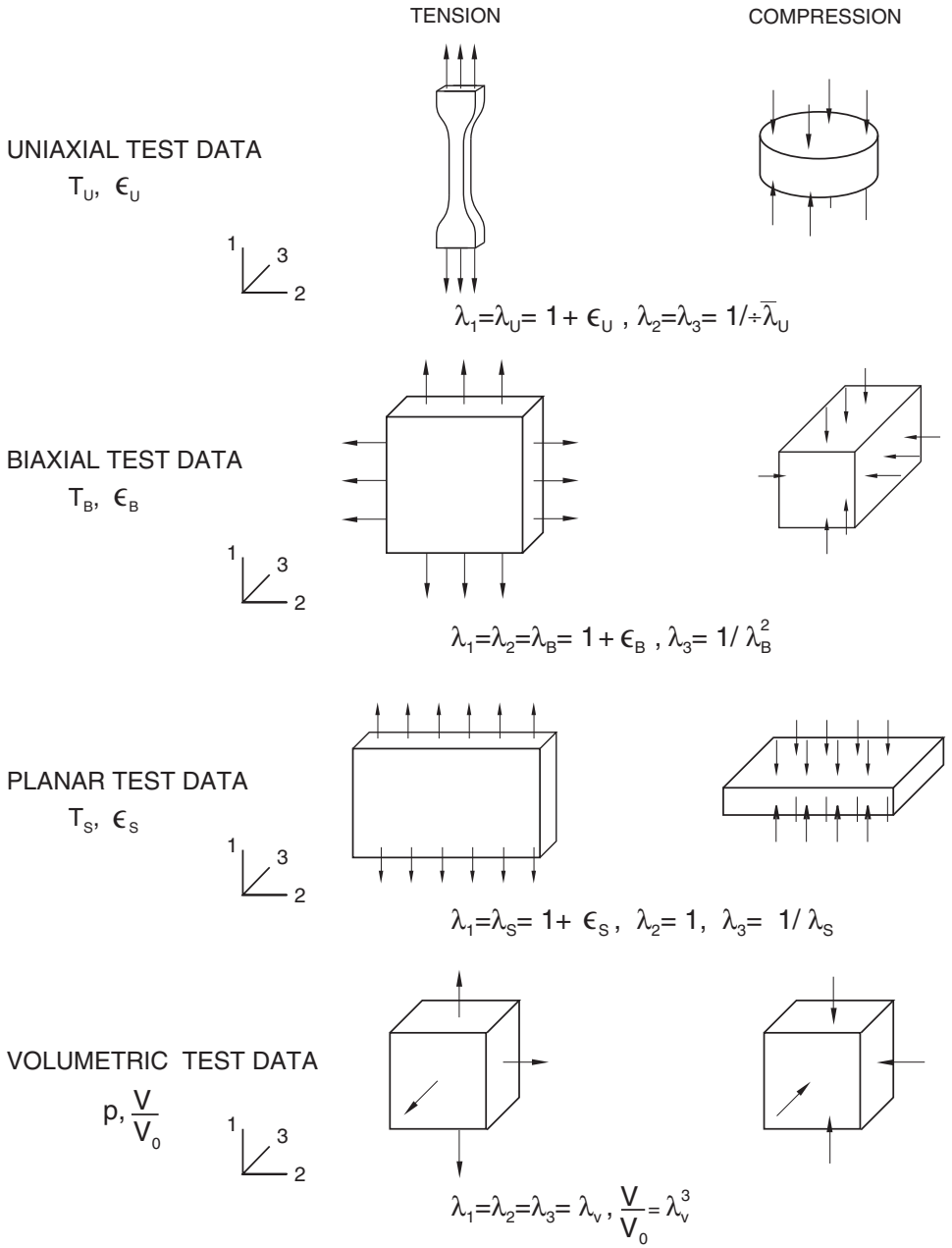


Figure 22.5.1–2 Schematic illustrations of deformation modes.

Combine data from these three test types to get a good characterization of the hyperelastic material behavior.

For the incompressible version of the material model, the stress-strain relationships for the different tests are developed using derivatives of the strain energy function with respect to the strain invariants. We define these relations in terms of the nominal stress (the force divided by the original, undeformed area) and the nominal, or engineering, strain defined below.

The deformation gradient, expressed in the principal directions of stretch, is

$$\mathbf{F} = \begin{bmatrix} \lambda_1 & 0 & \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},$$

where λ_1, λ_2, and λ_3 are the principal stretches: the ratios of current length to length in the original configuration in the principal directions of a material fiber. The principal stretches, λ_i, are related to the principal nominal strains, ϵ_i, by

$$\lambda_i = 1 + \epsilon_i.$$

Because we assume incompressibility and isothermal response, $J = \det(\mathbf{F}) = 1$ and, hence, $\lambda_1\lambda_2\lambda_3 = 1$. The deviatoric strain invariants in terms of the principal stretches are then

$$\overline{I}_1 = {\lambda_1}^2 + {\lambda_2}^2 + {\lambda_3}^2,$$

and

$$\overline{I}_2 = {\lambda_1}^{-2} + {\lambda_2}^{-2} + {\lambda_3}^{-2}.$$

Uniaxial tests

The uniaxial deformation mode is characterized in terms of the principal stretches, λ_i, as

$$\lambda_1 = \lambda_U, \quad \lambda_2 = \lambda_3 = 1/\sqrt{\lambda_U},$$

where λ_U is the stretch in the loading direction. The nominal strain is defined by $\varepsilon_U = \lambda_U - 1$.

To derive the uniaxial nominal stress T_U, we invoke the principle of virtual work:

$$\delta U = T_U \delta\lambda_U,$$

so that

$$T_U = \frac{\partial U}{\partial \lambda_U} = 2(1 - \lambda_U^{-3}) \left(\lambda_U \frac{\partial U}{\partial \overline{I}_1} + \frac{\partial U}{\partial \overline{I}_2} \right).$$

The uniaxial tension test is the most common of all the tests and is usually performed by pulling a "dog-bone" specimen. The uniaxial compression test is performed by loading a compression button between lubricated surfaces. The loading surfaces are lubricated to minimize any barreling effect in the button that would cause deviations from a homogeneous uniaxial compression stress-strain state.

Input File Usage: *UNIAXIAL TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data→Uniaxial Test Data**

Equibiaxial tests

The equibiaxial deformation mode is characterized in terms of the principal stretches, λ_i, as

$$\lambda_1 = \lambda_2 = \lambda_B, \quad \lambda_3 = 1/\lambda_B^2,$$

where λ_B is the stretch in the two perpendicular loading directions. The nominal strain is defined by $\varepsilon_B = \lambda_B - 1$.

To develop the expression for the equibiaxial nominal stress, T_B, we again use the principle of virtual work (assuming that the stress perpendicular to the loading direction is zero),

$$\delta U = 2T_B \delta\lambda_B,$$

so that

$$T_B = \frac{1}{2}\frac{\partial U}{\partial \lambda_B} = 2\left(\lambda_B - \lambda_B^{-5}\right)\left(\frac{\partial U}{\partial \bar{I}_1} + \lambda_B^2 \frac{\partial U}{\partial \bar{I}_2}\right).$$

In practice, the equibiaxial compression test is rarely performed because of experimental setup difficulties. In addition, this deformation mode is equivalent to a uniaxial tension test, which is straightforward to conduct.

A more common test is the equibiaxial tension test, in which a stress state with two equal tensile stresses and zero shear stress is created. This state is usually achieved by stretching a square sheet in a biaxial testing machine. It can also be obtained by inflating a circular membrane into a spheroidal shape (like blowing up a balloon). The stress field in the middle of the membrane then closely approximates equibiaxial tension, provided that the thickness of the membrane is very much smaller than the radius of curvature at this point. However, the strain distribution will not be quite uniform, and local strain measurements will be required. Once the strain and radius of curvature are known, the nominal stress can be derived from the inflation pressure.

Input File Usage: *BIAXIAL TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data→Biaxial Test Data**

Planar tests

The planar deformation mode is characterized in terms of the principal stretches, λ_i, as

$$\lambda_1 = \lambda_S,\ \lambda_2 = 1,\ \lambda_3 = 1/\lambda_S,$$

where λ_S is the stretch in the loading direction. Then, the nominal strain in the loading direction is $\varepsilon_S = \lambda_S - 1$.

This test is also called a "pure shear" test since, in terms of logarithmic strains,

$$\varepsilon_1 = \ln \lambda_1 = -\ln \lambda_3 = -\varepsilon_3, \quad \varepsilon_2 = \ln \lambda_2 = 0,$$

which corresponds to a state of pure shear at an angle of 45° to the loading direction.

The principle of virtual work gives

$$\delta U = T_S \delta \lambda_S,$$

where T_S is the nominal planar stress, so that

$$T_S = \frac{\partial U}{\partial \lambda_S} = 2(\lambda_S - \lambda_S^{-3}) \left(\frac{\partial U}{\partial \overline{I}_1} + \frac{\partial U}{\partial \overline{I}_2} \right).$$

For the (general) polynomial and Ogden models and for the coefficient β in the Van der Waals model this equation alone will not determine the constants uniquely. The planar test data must be augmented by uniaxial test data and/or biaxial test data to determine the material parameters.

Planar tests are usually done with a thin, short, and wide rectangular strip of material fixed on its wide edges to rigid loading clamps that are moved apart. If the separation direction is the *1*-direction and the thickness direction is the *3*-direction, the comparatively long size of the specimen in the *2*-direction and the rigid clamps allow us to use the approximation $\lambda_2 = 1$; that is, there is no deformation in the wide direction of the specimen. This deformation mode could also be called planar compression if the *3*-direction is considered to be the primary direction. All forms of incompressible plane strain behavior are characterized by this deformation mode. Consequently, if plane strain analysis is performed, planar test data represent the relevant form of straining of the material.

Input File Usage: *PLANAR TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data→Planar Test Data**

Volumetric tests

The following discussion describes procedures for obtaining D_i values (or D, for the Arruda-Boyce and Van der Waals models) corresponding to the actual material behavior. With these values you can compare the material's initial bulk modulus, $K_0 = 2/D_1$, to its initial shear modulus ($\mu_0 = 2(C_{10} + C_{01})$ for the polynomial model, $\mu_0 = \sum_{i=1}^{N} \mu_i$ for Ogden's model) and then judge whether D_i values that will

provide results are sufficiently realistic. For Abaqus/Explicit caution should be used; K_0/μ_0 should be less than 100. Otherwise, noisy solutions will be obtained and time increments will be excessively small (see "Compressibility in Abaqus/Explicit" above). The D_i and $\boldsymbol{D}$ can be calculated from data obtained in pure volumetric compression of a specimen (volumetric tension tests are much more difficult to perform). In a pure volumetric test $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V$; therefore, $\overline{I}_1 = \overline{I}_2 = 3$ and $J = {\lambda_V}^3 = V/V_0$ (the volume ratio). Using the polynomial form of the strain energy potential, the total pressure stress on the specimen is obtained as

$$p = -\left(\frac{\sigma_1 + \sigma_2 + \sigma_3}{3}\right) = -\sum_{i=1}^{N} 2i\frac{1}{D_i}({\lambda_V}^3 - 1)^{2i-1}.$$

This equation can be used to determine the D_i. If we are using a second-order polynomial series for $\boldsymbol{U}$, we have $N = 2$, and so two D_i are needed. Therefore, a minimum of two points on the pressure-volume ratio curve are required to give two equations for the D_i. For the Ogden and reduced polynomial potentials D_i can be determined for up to $N = 6$. A linear least-squares fit is performed when more than $\boldsymbol{N}$ data points are provided.

An approximate way of conducting a volumetric test consists of using a cylindrical rubber specimen that fits snugly inside a rigid container and whose top surface is compressed by a rigid piston. Although both volumetric and deviatoric deformation are present, the deviatoric stresses will be several orders of magnitude smaller than the hydrostatic stresses (because the bulk modulus is much higher than the shear modulus) and can be neglected. The compressive stress imposed by the rigid piston is effectively the pressure, and the volumetric strain in the rubber cylinder is computed from the piston displacement.

Nonzero values of D_i affect the uniaxial, equibiaxial, and planar stress results. However, since the material is assumed to be only slightly compressible, the techniques described for obtaining the deviatoric coefficients should give sufficiently accurate values even though they assume that the material is fully incompressible.

Input File Usage: *VOLUMETRIC TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data→Volumetric Test Data**

Equivalent experimental tests

The superposition of a tensile or compressive hydrostatic stress on a loaded, fully incompressible elastic body results in different stresses but does not change the deformation. Thus, Figure 22.5.1–3 shows that some apparently different loading conditions are actually equivalent in their deformations and, therefore, are equivalent tests:

- Uniaxial tension $\Longleftrightarrow$ Equibiaxial compression
- Uniaxial compression $\Longleftrightarrow$ Equibiaxial tension
- Planar tension $\Longleftrightarrow$ Planar compression

On the other hand, the tensile and compressive cases of the uniaxial and equibiaxial modes are independent from each other: uniaxial tension and uniaxial compression provide independent data.

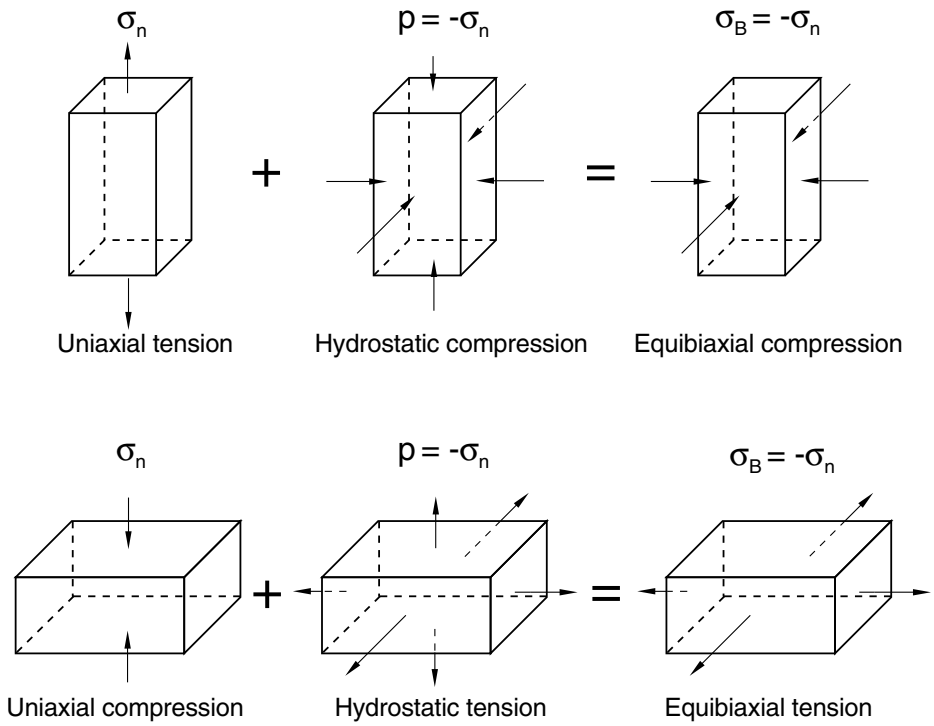


Figure 22.5.1–3 Equivalent deformation modes through superposition of hydrostatic stress.

Smoothing the test data

Experimental test data often contain noise in the sense that the test variable is both slowly varying and also corrupted by random noise. This noise can affect the quality of the strain energy potential that Abaqus derives. This noise is particularly a problem with the Marlow form, where a strain energy potential that exactly describes the test data that are used to calibrate the model is computed. It is less of a concern with the other forms, since smooth functions are fitted through the test data.

Abaqus provides a smoothing technique to remove the noise from the test data based on the Savitzky-Golay method. The idea is to replace each data point by a local average of its surrounding data points, so that the level of noise can be reduced without biasing the dominant trend of the test data. In the implementation a cubic polynomial is fitted through each data point i and n data points to the immediate left and right of that point. A least-squares method is used to fit the polynomial through these

$2n + 1$ points. The value of data point i is then replaced by the value of the polynomial at the same position. Each polynomial is used to adjust one data point except near the ends of the curve, where a polynomial is used to adjust multiple points, because the first and last few points cannot be the center of the fitting set of data points. This process is applied repeatedly to all data points until two consecutive passes through the data produce nearly the same results.

By default, the test data are not smoothed. If smoothing is specified, the default value is n=3. Alternatively, you can specify the number of data points to the left and right of a data point in the moving window within which a least-squares polynomial is fit.

Input File Usage: For the Marlow form, use one of the first three options and, optionally, the fourth option; for the other potential forms, use one and up to four of the following options:

*UNIAXIAL TEST DATA, SMOOTH=*n* ($n \geq 2$)
*BIAXIAL TEST DATA, SMOOTH=*n* ($n \geq 2$)
*PLANAR TEST DATA, SMOOTH=*n* ($n \geq 2$)
*VOLUMETRIC TEST DATA, SMOOTH=*n* ($n \geq 2$)

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data→Uniaxial Test Data**, **Biaxial Test Data**, **Planar Test Data**, or **Volumetric Test Data**

In each of the **Test Data Editor** dialog boxes, toggle on **Apply smoothing**, and select a value for *n* ($n \geq 2$).

Model prediction of material behavior versus experimental data

Once the strain energy potential is determined, the behavior of the hyperelastic model in Abaqus is established. However, the quality of this behavior must be assessed: the prediction of material behavior under different deformation modes must be compared against the experimental data. You must judge whether the strain energy potentials determined by Abaqus are acceptable, based on the correlation between the Abaqus predictions and the experimental data. You can evaluate the hyperelastic behavior automatically in Abaqus/CAE. Alternatively, single-element test cases can be used to derive the nominal stress–nominal strain response of the material model.Single-element test cases can be used to derive the nominal stress–nominal strain response of the material model.

See "Fitting of rubber test data," Section 3.1.4 of the Abaqus Benchmarks Manual, which illustrates the entire process of fitting hyperelastic constants to a set of test data.

Hyperelastic material stability

An important consideration in judging the quality of the fit to experimental data is the concept of material or Drucker stability. Abaqus checks the Drucker stability of the material for the first three deformation modes described above.

The Drucker stability condition for an incompressible material requires that the change in the stress, $d\boldsymbol{\sigma}$, following from any infinitesimal change in the logarithmic strain, $d\boldsymbol{\varepsilon}$, satisfies the inequality

$$d\boldsymbol{\sigma} : d\boldsymbol{\varepsilon} > 0.$$

Using $d\boldsymbol{\sigma} = \mathbf{D} : d\boldsymbol{\varepsilon}$, where $\mathbf{D}$ is the tangent material stiffness, the inequality becomes

$$d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0,$$

thus requiring the tangential material stiffness to be positive-definite.

For an isotropic elastic formulation the inequality can be represented in terms of the principal stresses and strains,

$$d\sigma_1 d\varepsilon_1 + d\sigma_2 d\varepsilon_2 + d\sigma_3 d\varepsilon_3 > 0.$$

As before, since the material is assumed to be incompressible, we can choose any value for the hydrostatic pressure without affecting the strains. A convenient choice for the stability calculation is $\sigma_3 = d\sigma_3 = 0$, which allows us to ignore the third term in the above equation.

The relation between the changes in stress and in strain can then be obtained in the form of the matrix

$$\begin{pmatrix} d\sigma_1 \\ d\sigma_2 \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} \begin{pmatrix} d\varepsilon_1 \\ d\varepsilon_2 \end{pmatrix},$$

where $D_{ij} = D_{ij}(\lambda_1, \lambda_2, \lambda_3)$. For material stability $\mathbf{D}$ must be positive-definite; thus, it is necessary that

$$D_{11} + D_{22} > 0,$$

$$D_{11} D_{22} - D_{12} D_{21} > 0.$$

This stability check is performed for the polynomial models, the Ogden potential, the Van der Waals form, and the Marlow form. The Arruda-Boyce form is always stable for positive values of (μ, λ_m); hence, it suffices to check the material coefficients to ensure stability.

You should be careful when defining the C_{ij} or (μ_i, α_i) for the polynomial models or the Ogden form: especially when $N > 1$, the behavior at higher strains is strongly sensitive to the values of the C_{ij} or (μ_i, α_i), and unstable material behavior may result if these values are not defined correctly. When some of the coefficients are strongly negative, instability at higher strain levels is likely to occur.

Abaqus performs a check on the stability of the material for six different forms of loading—uniaxial tension and compression, equibiaxial tension and compression, and planar tension and compression—for $0.1 \leq \lambda_1 \leq 10.0$ (nominal strain range of $-0.9 \leq \epsilon_1 \leq 9.0$) at intervals $\Delta\lambda_1 = 0.01$. If an instability is found, Abaqus issues a warning message and prints the lowest absolute value of ϵ_1 for which the instability is observed. Ideally, no instability occurs. If instabilities are observed at strain levels that are likely to occur in the analysis, it is strongly recommended that you either change the material model or

carefully examine and revise the material input data. If user subroutine **UHYPER** is used to define the hyperelastic material, you are responsible for ensuring stability.

Improving the accuracy and stability of the test data fit

Unfortunately, the initial fit of the models to experimental data may not come out as well as expected. This is particularly true for the most general models, such as the (general) polynomial model and the Ogden model. For some of the simpler models, stability is assured by following some simple rules.

- For positive values of the initial shear modulus, μ, and the locking stretch, λ_m, the Arruda-Boyce form is always stable.
- For positive values of the coefficient C_{10} the neo-Hookean form is always stable.
- Given positive values of the initial shear modulus, μ, and the locking stretch, λ_m, the stability of the Van der Waals model depends on the global interaction parameter, $\boldsymbol{a}$.
- For the Yeoh model stability is assured if all $C_{i0} > 0$. Typically, however, C_{20} will be negative, since this helps capture the S-shape feature of the stress-strain curve. Thus, reducing the absolute value of C_{20} or magnifying the absolute value of C_{10} will help make the Yeoh model more stable.

In all cases the following suggestions may improve the quality of the fit:

- Both tension and compression data are allowed; compressive stresses and strains are entered as negative values. Use compression or tension data depending on the application: it is difficult to fit a single material model accurately to both tensile and compressive data.
- Always use many more experimental data points than unknown coefficients.
- If $N \geq 3$ is used, experimental data should be available to at least 100% tensile strain or 50% compressive strain.
- Perform different types of tests (e.g., compression and simple shear tests). Proper material behavior for a deformation mode requires test data to characterize that mode.
- Check for warning messages about material instability or error messages about lack of convergence in fitting the test data. This check is especially important with new test data; a simple finite element model with the new test data can be run through the analysis input file processor to check the material stability.
- Use the material evaluation capability in Abaqus/CAE to compare the response curves for different strain energy potentials to the experimental data. Alternatively, you can perform one-element simulations for simple deformation modes and compare the Abaqus results against the experimental data. The *X–Y* plotting options in the Visualization module of Abaqus/CAE can be used for this comparison.

  You can perform one-element simulations for simple deformation modes and compare the Abaqus results against the experimental data.
- Delete some data points at very low strains if large strains are anticipated. A disproportionate number of low strain points may unnecessarily bias the accuracy of the fit toward the low strain range and cause greater errors in the large strain range.

- Delete some data points at the highest strains if small to moderate strains are expected. The high strain points may force the fitting to lose accuracy and/or stability in the low strain range.
- Pick data points at evenly spaced strain intervals over the expected range of strains, which will result in similar accuracy throughout the entire strain range.
- The higher the order of N, the more oscillations are likely to occur, leading to instabilities in the stress-strain curves. If the (general) polynomial model is used, lower the order of N from 2 to 1 (3 to 2 for Ogden), especially if the maximum strain level is low (say, less than 100% strain).
- If multiple types of test data are used and the fit still comes out poorly, some of the test data probably contain experimental errors. New tests may be needed. One way of determining which test data are erroneous is to first calibrate the initial shear modulus μ_0^{test} of the material. Then fit each type of test data separately in Abaqus and compute the shear modulus, μ_0^{fit}, from the material constants using the relations

$$\mu_0^{fit} = 2(C_{10} + C_{01}) \quad \text{(polynomial form)} \qquad \text{or} \qquad \mu_0^{fit} = \sum_{i=1}^{N} \mu_i \quad \text{(Ogden form)}.$$

Alternatively, the initial Young's modulus, E_0^{test}, can be calibrated and compared with

$$E_0^{fit} = 6(C_{10} + C_{01}) \quad \text{(polynomial form)} \qquad \text{or} \qquad E_0^{fit} = 3\sum_{i=1}^{N} \mu_i \quad \text{(Ogden form)}.$$

The values of μ_0^{fit} or E_0^{fit} that are most different from μ_0^{test} or E_0^{test} indicate the erroneous test data.

Elements

The hyperelastic material model can be used with solid (continuum) elements, finite-strain shells (except S4), continuum shells, membranes, and one-dimensional elements (trusses and rebars). In Abaqus/Standard the hyperelastic material model can be also used with Timoshenko beams (B21, B22, B31, B31OS, B32, B32OS, PIPE21, PIPE22, PIPE31, PIPE32, and their "hybrid" equivalents). It cannot be used with Euler-Bernoulli beams (B23, B23H, B33, and B33H) and small-strain shells (STRI3, STRI65, S4R5, S8R, S8R5, S9R5).

Pure displacement formulation versus hybrid formulation in Abaqus/Standard

For continuum elements in Abaqus/Standard hyperelasticity can be used with the pure displacement formulation elements or with the "hybrid" (mixed formulation) elements. Because elastomeric materials are usually almost incompressible, fully integrated pure displacement method elements are not recommended for use with this material, except for plane stress cases. If fully or selectively reduced-integration displacement method elements are used with the almost incompressible form of this material model, a penalty method is used to impose the incompressibility constraint in anything except plane stress analysis. The penalty method can sometimes lead to numerical difficulties; therefore, the

fully or selectively reduced-integrated "hybrid" formulation elements are recommended for use with hyperelastic materials.

In general, an analysis using a single hybrid element will be only slightly more computationally expensive than an analysis using a regular displacement-based element. However, when the wavefront is optimized, the Lagrange multipliers may not be ordered independently of the regular degrees of freedom associated with the element. Thus, the wavefront of a very large mesh of second-order hybrid tetrahedra may be noticeably larger than that of an equivalent mesh using regular second-order tetrahedra. This may lead to significantly higher CPU costs, disk space, and memory requirements.

Incompatible mode elements in Abaqus/Standard

Incompatible mode elements should be used with caution in applications involving large strains. Convergence may be slow, and in hyperelastic applications inaccuracies may accumulate. Erroneous stresses may sometimes appear in incompatible mode hyperelastic elements that are unloaded after having been subjected to a complex deformation history.

Procedures

Hyperelasticity must always be used with geometrically nonlinear analyses ("General and linear perturbation procedures," Section 6.1.3).

22.5.2 HYPERELASTIC BEHAVIOR IN ELASTOMERIC FOAMS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Energy dissipation in elastomeric foams," Section 22.6.2
- *HYPERFOAM
- *UNIAXIAL TEST DATA
- *BIAXIAL TEST DATA
- *PLANAR TEST DATA
- *VOLUMETRIC TEST DATA
- *SIMPLE SHEAR TEST DATA
- *MULLINS EFFECT
- "Creating a hyperfoam material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The elastomeric foam material model:

- is isotropic and nonlinear;
- is valid for cellular solids whose porosity permits very large volumetric changes;
- optionally allows the specification of energy dissipation and stress softening effects (see "Energy dissipation in elastomeric foams," Section 22.6.2);
- can deform elastically to large strains, up to 90% strain in compression; and
- requires that geometric nonlinearity be accounted for during the analysis step (see "Defining an analysis," Section 6.1.2, and "General and linear perturbation procedures," Section 6.1.3), since it is intended for finite-strain applications.

Abaqus/Explicit also provides a separate foam material model intended to capture the strain-rate sensitive behavior of low-density elastomeric foams such as used in crash and impact applications (see "Low-density foams," Section 22.9.1).

Mechanical behavior of elastomeric foams

Cellular solids are made up of interconnected networks of solid struts or plates that form the edges and faces of cells. Foams are made up of polyhedral cells that pack in three dimensions. The foam cells can be either open (e.g., sponge) or closed (e.g., flotation foam). Common examples of elastomeric

foam materials are cellular polymers such as cushions, padding, and packaging materials that utilize the excellent energy absorption properties of foams: the energy absorbed by foams is substantially greater than that absorbed by ordinary stiff elastic materials for a certain stress level.

Another class of foam materials is crushable foams, which undergo permanent (plastic) deformation. Crushable foams are discussed in "Crushable foam plasticity models," Section 23.3.5.

Foams are commonly loaded in compression. Figure 22.5.2–1 shows a typical compressive stress-strain curve.

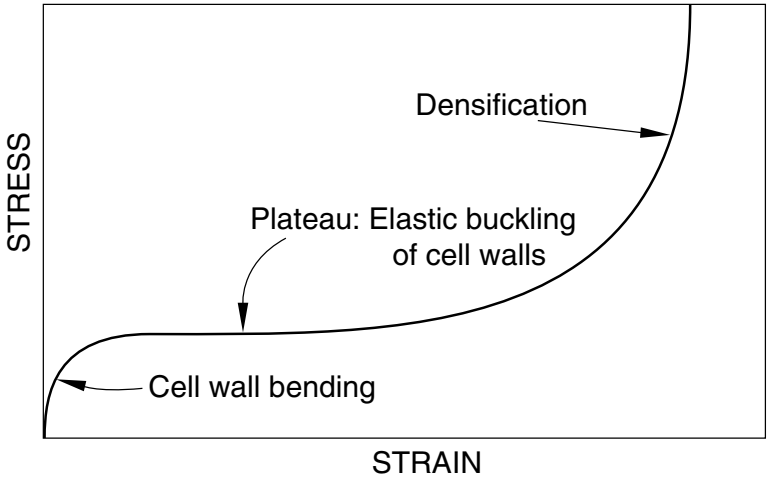


Figure 22.5.2–1 Typical compressive stress-strain curve.

Three stages can be distinguished during compression:

1. At small strains ($< 5\%$) the foam deforms in a linear elastic manner due to cell wall bending.
2. The next stage is a plateau of deformation at almost constant stress, caused by the elastic buckling of the columns or plates that make up the cell edges or walls. In closed cells the enclosed gas pressure and membrane stretching increase the level and slope of the plateau.
3. Finally, a region of densification occurs, where the cell walls crush together, resulting in a rapid increase of compressive stress. Ultimate compressive nominal strains of 0.7 to 0.9 are typical.

The tensile deformation mechanisms for small strains are similar to the compression mechanisms, but they differ for large strains. Figure 22.5.2–2 shows a typical tensile stress-strain curve. There are two stages during tension:

1. At small strains the foam deforms in a linear, elastic manner as a result of cell wall bending, similar to that in compression.
2. The cell walls rotate and align, resulting in rising stiffness. The walls are substantially aligned at a tensile strain of about $1/3$. Further stretching results in increased axial strains in the walls.

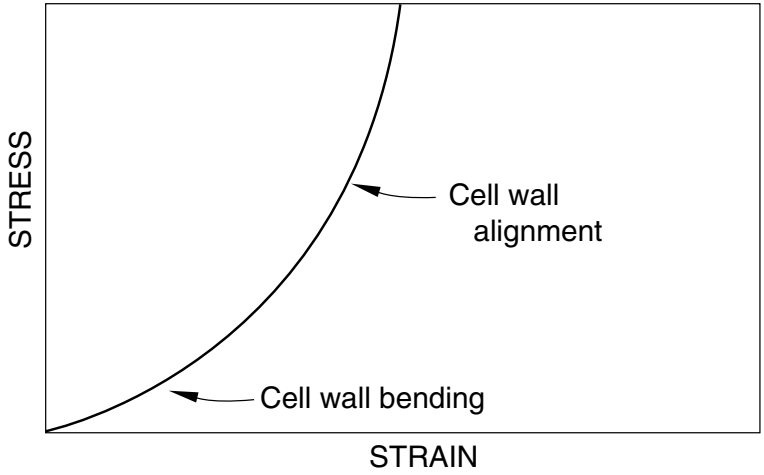


Figure 22.5.2–2 Typical tensile stress-strain curve.

At small strains for both compression and tension, the average experimentally observed Poisson's ratio, ν, of foams is 1/3. At larger strains it is commonly observed that Poisson's ratio is effectively zero during compression: the buckling of the cell walls does not result in any significant lateral deformation. However, ν is nonzero during tension, which is a result of the alignment and stretching of the cell walls.

The manufacture of foams often results in cells with different principal dimensions. This shape anisotropy results in different loading responses in different directions. However, the hyperfoam model does not take this kind of initial anisotropy into account.

Strain energy potential

In the elastomeric foam material model the elastic behavior of the foams is based on the strain energy function

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left[\hat{\lambda}_1^{\alpha_i} + \hat{\lambda}_2^{\alpha_i} + \hat{\lambda}_3^{\alpha_i} - 3 + \frac{1}{\beta_i} \left((J^{e\ell})^{-\alpha_i \beta_i} - 1 \right) \right],$$

where N is a material parameter; μ_i, α_i, and β_i are temperature-dependent material parameters;

$$\hat{\lambda}_i = (J^{th})^{-\frac{1}{3}} \lambda_i \quad \rightarrow \quad \hat{\lambda}_1 \hat{\lambda}_2 \hat{\lambda}_3 = J^{e\ell};$$

and λ_i are the principal stretches. The elastic and thermal volume ratios, $J^{e\ell}$ and J^{th}, are defined below.

The coefficients μ_i are related to the initial shear modulus, μ_0, by

$$\mu_0 = \sum_{i=1}^{N} \mu_i,$$

while the initial bulk modulus, K_0, follows from

$$K_0 = \sum_{i=1}^{N} 2\mu_i(\frac{1}{3} + \beta_i).$$

For each term in the energy function, the coefficient β_i determines the degree of compressibility. β_i is related to the Poisson's ratio, ν_i, by the expressions

$$\beta_i = \frac{\nu_i}{1-2\nu_i}, \qquad \nu_i = \frac{\beta_i}{1+2\beta_i}.$$

Thus, if β_i is the same for all terms, we have a single effective Poisson's ratio, ν. This effective Poisson's ratio is valid for finite values of the logarithmic principal strains $\varepsilon_1, \varepsilon_2, \varepsilon_3$; in uniaxial tension $\varepsilon_2 = \varepsilon_3 = -\nu\varepsilon_1$.

Thermal expansion

Only isotropic thermal expansion is permitted with the hyperfoam material model.

The elastic volume ratio, $J^{e\ell}$, relates the total volume ratio (current volume/reference volume), J, and the thermal volume ratio, J^{th}:

$$J^{e\ell} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = (1 + \varepsilon^{th})^3,$$

where ε^{th} is the linear thermal expansion strain that is obtained from the temperature and the isotropic thermal expansion coefficient ("Thermal expansion," Section 26.1.2).

Determining the hyperfoam material parameters

The response of the material is defined by the parameters in the strain energy function, U; these parameters must be determined to use the hyperfoam model. Two methods are provided for defining the material parameters: you can specify the hyperfoam material parameters directly or specify test data and allow Abaqus to calculate the material parameters.

The elastic response of a viscoelastic material ("Time domain viscoelasticity," Section 22.7.1) can be specified by defining either the instantaneous response or the long-term response of such a material. To define the instantaneous response, the experiments outlined in the "Experimental tests" section that

follows have to be performed within time spans much shorter than the characteristic relaxation time of the material.

Input File Usage: *HYPERFOAM, MODULI=INSTANTANEOUS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: **Moduli time scale (for viscoelasticity): Instantaneous**

If, on the other hand, the long-term elastic response is used, data from experiments have to be collected after time spans much longer than the characteristic relaxation time of the viscoelastic material. Long-term elastic response is the default elastic material behavior.

Input File Usage: *HYPERFOAM, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: **Moduli time scale (for viscoelasticity): Long-term**

Direct specification

When the parameters N, μ_i, α_i, and ν_i are specified directly, they can be functions of temperature.

The default value of ν_i is zero, which corresponds to an effective Poisson's ratio of zero. The incompressible limit corresponds to all $\nu_i \rightarrow 0.5$. However, this material model should not be used for approximately incompressible materials: use of the hyperelastic model ("Hyperelastic behavior of rubberlike materials," Section 22.5.1) is recommended if the effective Poisson's ratio $\nu_{eff} > 0.45$.

Input File Usage: *HYPERFOAM, N=*n* ($n \leq 6$)

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: **Strain energy potential order: *n*** ($n \leq 6$); optionally, toggle on **Use temperature-dependent data**

Test data specification

The value of N and the experimental stress-strain data can be specified for up to five simple tests: uniaxial, equibiaxial, simple shear, planar, and volumetric. Abaqus contains a capability for obtaining the μ_i, α_i, and β_i for the hyperfoam model with up to six terms (N=6) directly from test data. Poisson effects can be included either by means of a constant Poisson's ratio or through specification of volumetric test data and/or lateral strains in the other test data.

It is important to recognize that the properties of foam materials can vary significantly from one batch to another. Therefore, all of the experiments should be performed on specimens taken from the same batch of material.

This method does not allow the properties to be temperature dependent.

As many data points as required can be entered from each test. Abaqus will then compute μ_i, α_i, and, if necessary, ν_i. The technique uses a least squares fit to the experimental data so that the relative error in the nominal stress is minimized.

It is recommended that data from the uniaxial, biaxial, and simple shear tests (on samples taken from the same piece of material) be included and that the data points cover the range of nominal strains expected to arise in the actual loading. The planar and volumetric tests are optional.

For all tests the strain data, including the lateral strain data, should be given as nominal strain values (change in length per unit of original length). For the uniaxial, equibiaxial, simple shear, and planar tests, stress data are given as nominal stress values (force per unit of original cross-sectional area). The tests allow for both compression and tension data; compressive stresses and strains should be entered as negative values. For the volumetric tests the stress data are given as pressure values.

Input File Usage: Use the first option to define an effective Poisson's ratio ($\nu_i = \nu$ for all i), or use the second option to define the lateral strains as part of the test data input:

*HYPERFOAM, N=*n*, POISSON=ν, TEST DATA INPUT ($n \leq 6$)
*HYPERFOAM, N=*n*, TEST DATA INPUT ($n \leq 6$).

In addition, use at least one and up to five of these additional options to give the experimental stress-strain data (see "Experimental tests" below):

*UNIAXIAL TEST DATA
*BIAXIAL TEST DATA
*PLANAR TEST DATA
*SIMPLE SHEAR TEST DATA
*VOLUMETRIC TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: toggle on **Use test data**; **Strain energy potential order: *n*** ($n \leq 6$); optionally, toggle on **Use constant Poisson's ratio:** and enter a value for the effective Poisson's ratio ($\nu_i = \nu$ for all i)

In addition, use at least one and up to five of the suboptions to give the experimental stress-strain data (see "Experimental tests" below):

Suboptions→Uniaxial Test Data

Suboptions→Biaxial Test Data

Suboptions→Planar Test Data

Suboptions→Simple Shear Test Data

Suboptions→Volumetric Test Data

Experimental tests

For a homogeneous material, homogeneous deformation modes suffice to characterize the material parameters. Abaqus accepts test data from the following deformation modes:

- Uniaxial tension and compression
- Equibiaxial tension and compression
- Planar tension and compression (pure shear)
- Simple shear
- Volumetric tension and compression

The stress-strain relations are defined in terms of the nominal stress (the force divided by the original, undeformed area) and the nominal, or engineering, strains, ϵ_i. The principal stretches, λ_i, are related to the principal nominal strains, ϵ_i, by

$$\lambda_i = 1 + \epsilon_i.$$

Uniaxial, equibiaxial, and planar tests

The deformation gradient, expressed in the principal directions of stretch, is

$$\mathbf{F} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},$$

where λ_1, λ_2, and λ_3 are the principal stretches: the ratios of current length to length in the original configuration in the principal directions of a material fiber. The deformation modes are characterized in terms of the principal stretches, λ_i, and the volume ratio, $J = \det(\mathbf{F})$. The elastomeric foams are not incompressible, so that $J = \lambda_1\lambda_2\lambda_3 \neq 1$. The transverse stretches, λ_2 and/or λ_3, are independently specified in the test data either as individual values from the measured lateral deformations or through the definition of an effective Poisson's ratio.

The three deformation modes use a single form of the nominal stress-stretch relation,

$$T_L = \frac{\partial U}{\partial \lambda_L} = \frac{2}{\lambda_L} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\lambda_L^{\alpha_i} - J^{-\alpha_i \beta_i}),$$

where T_L is the nominal stress and λ_L is the stretch in the loading direction. Because of the compressible behavior, the planar mode does not result in a state of pure shear. In fact, if the effective Poisson's ratio is zero, planar deformation is identical to uniaxial deformation.

Uniaxial mode

In uniaxial mode $\lambda_1 = \lambda_U$, $\lambda_2 = \lambda_3$, and $J = \lambda_U \lambda_2^2$.

Input File Usage: *UNIAXIAL TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: toggle on **Use test data**, **Suboptions→Uniaxial Test Data**

Equibiaxial mode

In equibiaxial mode $\lambda_1 = \lambda_2 = \lambda_B$ and $J = \lambda_B^2 \lambda_3$.

Input File Usage: *BIAXIAL TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: toggle on **Use test data**, **Suboptions→Biaxial Test Data**

Planar mode

In planar mode $\lambda_1 = \lambda_P$, $\lambda_2 = 1$, and $J = \lambda_P \lambda_3$. Planar test data must be augmented by either uniaxial or biaxial test data.

Input File Usage: *PLANAR TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: toggle on **Use test data**, **Suboptions→Planar Test Data**

Simple shear tests

Simple shear is described by the deformation gradient

$$\mathbf{F} = \begin{bmatrix} 1 & \gamma & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where γ is the shear strain. For this deformation $J = \det(\mathbf{F}) = 1$. A schematic illustration of simple shear deformation is shown in Figure 22.5.2–3.

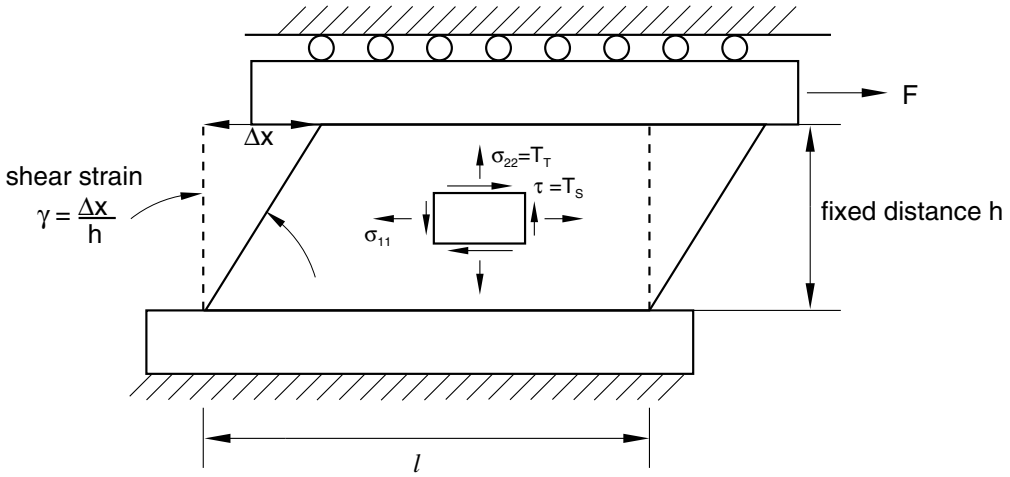


Figure 22.5.2–3 Simple shear test.

The nominal shear stress, T_S, is

$$T_S = \frac{\partial U}{\partial \gamma} = \sum_{j=1}^{2} \left\{ \frac{2\gamma}{2(\lambda_j^2 - 1) - \gamma^2} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\lambda_j^{\alpha_i} - 1) \right\},$$

where λ_j are the principal stretches in the plane of shearing, related to the shear strain γ by

$$\lambda_{1,2} = \sqrt{1 + \frac{\gamma^2}{2} \pm \gamma\sqrt{1 + \frac{\gamma^2}{4}}}.$$

The stretch in the direction perpendicular to the shear plane is $\lambda_3 = 1$. The transverse (tensile) stress, T_T, developed during simple shear deformation due to the Poynting effect, is

$$T_T = \frac{\partial U}{\partial \varepsilon} = \sum_{j=1}^{2} \left\{ \frac{2(\lambda_j^2 - 1)}{2\lambda_j^4 - \lambda_j^2(\gamma^2 + 2)} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\lambda_j^{\alpha_i} - 1) \right\}.$$

Input File Usage: *SIMPLE SHEAR TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: toggle on **Use test data**, **Suboptions→Simple Shear Test Data**

Volumetric tests

The deformation gradient, $\mathbf{F}$, is the same for volumetric tests as for uniaxial tests. The volumetric deformation mode consists of all principal stretches being equal;

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V, \; J = \lambda_V^3.$$

The pressure-volumetric ratio relation is

$$-p = \frac{\partial U}{\partial J} = \frac{2}{J} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (J^{\frac{1}{3}\alpha_i} - J^{-\alpha_i \beta_i}).$$

A volumetric compression test is illustrated in Figure 22.5.2–4. The pressure exerted on the foam specimen is the hydrostatic pressure of the fluid, and the decrease in the specimen volume is equal to the additional fluid entering the pressure chamber. The specimen is sealed against fluid penetration.

Input File Usage: *VOLUMETRIC TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperfoam**: toggle on **Use test data**, **Suboptions→Volumetric Test Data**

Difference between compression and tension deformation

For small strains ($< 5\%$) foams behave similarly for both compression and tension. However, at large strains the deformation mechanisms differ for compression (buckling and crushing) and tension (alignment and stretching). Therefore, accurate hyperfoam modeling requires that the experimental data used to define the material parameters correspond to the dominant deformation modes of the problem being analyzed. If compression dominates, the pertinent tests are:

- Uniaxial compression
- Simple shear
- Planar compression (if Poisson's ratio $\nu \neq 0$)

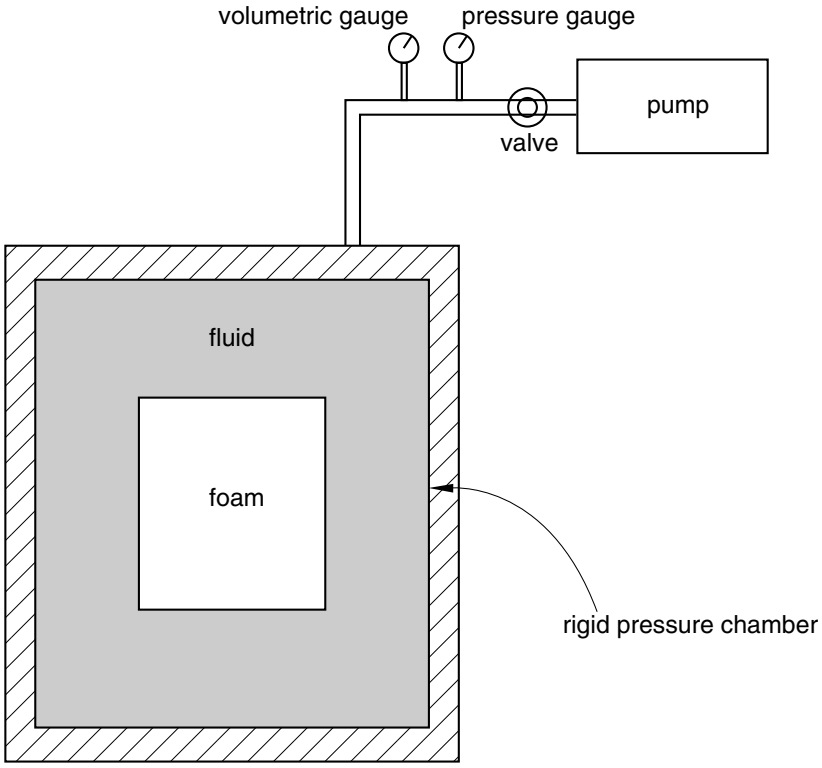


Figure 22.5.2–4 Volumetric compression test.

- Volumetric compression (if Poisson's ratio $\nu \neq 0$)

If tension dominates, the pertinent tests are:

- Uniaxial tension
- Simple shear
- Biaxial tension (if Poisson's ratio $\nu \neq 0$)
- Planar tension (if Poisson's ratio $\nu \neq 0$)

Lateral strain data can also be used to define the compressibility of the foam. Measurement of the lateral strains may make other tests redundant; for example, providing lateral strains for a uniaxial test eliminates the need for a volumetric test. However, if volumetric test data are provided in addition to the lateral strain data for other tests, both the volumetric test data and the lateral strain data will be used in determining the compressibility of the foam. The hyperfoam model may not accurately fit Poisson's ratio if it varies significantly between compression and tension.

Model prediction of material behavior versus experimental data

Once the elastomeric foam constants are determined, the behavior of the hyperfoam model in Abaqus is established. However, the quality of this behavior must be assessed: the prediction of material behavior under different deformation modes must be compared against the experimental data. You must judge whether the elastomeric foam constants determined by Abaqus are acceptable, based on the correlation between the Abaqus predictions and the experimental data. Single-element test cases can be used to calculate the nominal stress–nominal strain response of the material model.

See "Fitting of elastomeric foam test data," Section 3.1.5 of the Abaqus Benchmarks Manual, which illustrates the entire process of fitting elastomeric foam constants to a set of test data.

Elastomeric foam material stability

As with incompressible hyperelasticity, Abaqus checks the Drucker stability of the material for the deformation modes described above. The Drucker stability condition for a compressible material requires that the change in the Kirchhoff stress, $d\boldsymbol{\tau}$, following from an infinitesimal change in the logarithmic strain, $d\boldsymbol{\varepsilon}$, satisfies the inequality

$$d\boldsymbol{\tau} : d\boldsymbol{\varepsilon} > 0,$$

where the Kirchhoff stress $\boldsymbol{\tau} = J\boldsymbol{\sigma}$. Using $d\boldsymbol{\tau} = \mathbf{D} : d\boldsymbol{\varepsilon}$, the inequality becomes

$$d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0.$$

This restriction requires that the tangential material stiffness $\mathbf{D}$ be positive definite.

For an isotropic elastic formulation the inequality can be represented in terms of the principal stresses and strains

$$d\tau_1 d\varepsilon_1 + d\tau_2 d\varepsilon_2 + d\tau_3 d\varepsilon_3 > 0.$$

Thus, the relation between changes in the stress and changes in the strain can be obtained in the form of the matrix equation

$$\begin{Bmatrix} d\tau_1 \\ d\tau_2 \\ d\tau_3 \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \\ d\varepsilon_3 \end{Bmatrix},$$

where $D_{ij} = D_{ij}(\lambda_1, \lambda_2, \lambda_3)$.

Since $\mathbf{D}$ must be positive definite, it is necessary that

$$D_{11} + D_{22} + D_{33} > 0,$$

$$D_{11}D_{22} + D_{22}D_{33} + D_{33}D_{11} - D_{23}^2 - D_{13}^2 - D_{12}^2 > 0,$$

$$\det(\mathbf{D}) > 0.$$

You should be careful about defining the parameters μ_i, α_i, and ν_i: especially when $N > 1$, the behavior at higher strains is strongly sensitive to the values of these parameters, and unstable material behavior may result if these values are not defined correctly. When some of the coefficients are strongly negative, instability at higher strain levels is likely to occur. Abaqus performs a check on the stability of the material for nine different forms of loading—uniaxial tension and compression, equibiaxial tension and compression, simple shear, planar tension and compression, and volumetric tension and compression—for $0.1 \leq \lambda_1 \leq 10.0$ (nominal strain range of $-0.9 \leq \epsilon_1 \leq 9.0$), at intervals $\Delta\lambda_1 = 0.01$. If an instability is found, Abaqus issues a warning message and prints the lowest absolute value of ϵ_1 for which the instability is observed. Ideally, no instability occurs. If instabilities are observed at strain levels that are likely to occur in the analysis, it is strongly recommended that you carefully examine and revise the material input data.

Improving the accuracy and stability of the test data fit

"Hyperelastic behavior of rubberlike materials," Section 22.5.1, contains suggestions for improving the accuracy and stability of elastomeric modeling. "Fitting of elastomeric foam test data," Section 3.1.5 of the Abaqus Benchmarks Manual, illustrates the process of fitting elastomeric foam test data.

Elements

The hyperfoam model can be used with solid (continuum) elements, finite-strain shells (except S4), and membranes. However, it cannot be used with one-dimensional solid elements (trusses and beams), small-strain shells (STRI3, STRI65, S4R5, S8R, S8R5, S9R5), or the Eulerian elements (EC3D8R and EC3D8RT).

For continuum elements elastomeric foam hyperelasticity can be used with pure displacement formulation elements or, in Abaqus/Standard, with the "hybrid" (mixed formulation) elements. Since elastomeric foams are assumed to be very compressible, the use of hybrid elements will generally not yield any advantage over the use of purely displacement-based elements.

Procedures

The hyperfoam model must always be used with geometrically nonlinear analyses ("General and linear perturbation procedures," Section 6.1.3).

22.5.3 ANISOTROPIC HYPERELASTIC BEHAVIOR

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Mullins effect," Section 22.6.1
- *ANISOTROPIC HYPERELASTIC
- *VISCOELASTIC
- *MULLINS EFFECT
- "Creating an anisotropic hyperelastic material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The anisotropic hyperelastic material model:

- provides a general capability for modeling materials that exhibit highly anisotropic and nonlinear elastic behavior (such as biomedical soft tissues, fiber-reinforced elastomers, etc.);
- can be used in combination with large-strain time-domain viscoelasticity ("Time domain viscoelasticity," Section 22.7.1); however, viscoelasticity is isotropic;
- optionally allows the specification of energy dissipation and stress softening effects (see "Mullins effect," Section 22.6.1); and
- requires that geometric nonlinearity be accounted for during the analysis step ("General and linear perturbation procedures," Section 6.1.3) since it is intended for finite-strain applications.

Anisotropic hyperelasticity formulations

Many materials of industrial and technological interest exhibit anisotropic elastic behavior due to the presence of preferred directions in their microstructure. Examples of such materials include common engineering materials (such as fiber-reinforced composites, reinforced rubber, wood, etc.) as well as soft biological tissues (arterial walls, heart tissue, etc.). When these materials are subjected to small deformations (less than 2–5%), their mechanical behavior can generally be modeled adequately using conventional anisotropic linear elasticity (see "Defining fully anisotropic elasticity" in "Linear elastic behavior," Section 22.2.1). Under large deformations, however, these materials exhibit highly anisotropic and nonlinear elastic behavior due to rearrangements in the microstructure, such as reorientation of the fiber directions with deformation. The simulation of these nonlinear large-strain effects calls for more advanced constitutive models formulated within the framework of anisotropic hyperelasticity. Hyperelastic materials are described in terms of a "strain energy potential," U, which

defines the strain energy stored in the material per unit of reference volume (volume in the initial configuration) as a function of the deformation at that point in the material. Two distinct formulations are used for the representation of the strain energy potential of anisotropic hyperelastic materials: strain-based and invariant-based.

Strain-based formulation

In this case the strain energy function is expressed directly in terms of the components of a suitable strain tensor, such as the Green strain tensor (see "Strain measures," Section 1.4.2 of the Abaqus Theory Manual):

$$U = U(\boldsymbol{\varepsilon}^G),$$

where $\boldsymbol{\varepsilon}^G = \frac{1}{2}(\mathbf{C} - \mathbf{I})$ is Green's strain; $\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}$ is the right Cauchy-Green strain tensor; $\mathbf{F}$ is the deformation gradient; and $\mathbf{I}$ is the identity matrix. Without loss of generality, the strain energy function can be written in the form

$$U = U(\overline{\boldsymbol{\varepsilon}}^G, J^{e\ell}),$$

where $\overline{\boldsymbol{\varepsilon}}^G = \frac{1}{2}(\overline{\mathbf{C}} - \mathbf{I})$ is the modified Green strain tensor; $\overline{C} = J^{-\frac{2}{3}}\mathbf{C}$ is the distortional part of the right Cauchy-Green strain; $J = \det(\mathbf{F})$ is the total volume change; and $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion."

The underlying assumption in models based on the strain-based formulation is that the preferred material directions are initially aligned with an orthogonal coordinate system in the reference (stress-free) configuration. These directions may become non-orthogonal only after deformation. Examples of this form of strain energy function include the generalized Fung-type form described below.

Invariant-based formulation

Using the continuum theory of fiber-reinforced composites (Spencer, 1984) the strain energy function can be expressed directly in terms of the invariants of the deformation tensor and fiber directions. For example, consider a composite material that consists of an isotropic hyperelastic matrix reinforced with N families of fibers. The directions of the fibers in the reference configuration are characterized by a set of unit vectors $\mathbf{A}_\alpha$, $(\alpha = 1, ..., N)$. Assuming that the strain energy depends not only on deformation, but also on the fiber directions, the following form is postulated

$$U = U(\mathbf{C}, \mathbf{A}_\alpha); \qquad \alpha = 1, ..., N.$$

The strain energy of the material must remain unchanged if both matrix and fibers in the reference configuration undergo a rigid body rotation. Then, following Spencer (1984), the strain energy can be expressed in terms of an irreducible set of scalar invariants that form the integrity basis of the tensor $\mathbf{C}$ and the vectors $\mathbf{A}_\alpha$:

$$U = U\big(\overline{I}_1, \overline{I}_2, J^{e\ell}, \overline{I}_{4(\alpha\beta)}, \overline{I}_{5(\alpha\beta)}; \zeta_{\alpha\beta}\big); \qquad \alpha, \beta = 1, ..., N,$$

where $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants; $J^{e\ell}$ is the elastic volume ratio (or third strain invariant); $\overline{I}_{4(\alpha\beta)}$ and $\overline{I}_{5(\alpha\beta)}$ are the *pseudo-invariants* of $\overline{\mathbf{C}}$, $\mathbf{A}_\alpha$; and $\mathbf{A}_\beta$, defined as:

$$\overline{I}_{4(\alpha\beta)} = \mathbf{A}_\alpha \cdot \overline{\mathbf{C}} \cdot \mathbf{A}_\beta, \qquad \text{and} \qquad \overline{I}_{5(\alpha\beta)} = \mathbf{A}_\alpha \cdot \overline{\mathbf{C}}^2 \cdot \mathbf{A}_\beta; \qquad \alpha = 1, ..., N;\ \beta = 1, ..., \alpha.$$

The terms $\zeta_{\alpha\beta}$ are geometrical constants (independent of deformation) equal to the cosine of the angle between the directions of any two families of fibers in the reference configuration:

$$\zeta_{\alpha\beta} = \mathbf{A}_\alpha \cdot \mathbf{A}_\beta; \qquad \alpha = 1, ..., N;\ \beta = 1, ..., \alpha.$$

Unlike for the case of the strain-based formulation, in the invariant-based formulation the fiber directions need not be orthogonal in the initial configuration. An example of an invariant-based energy function is the form proposed by Holzapfel, Gasser, and Ogden (2000) for arterial walls (see "Holzapfel-Gasser-Ogden form," below).

Anisotropic strain energy potentials

There are two forms of strain energy potentials available in Abaqus to model approximately incompressible anisotropic materials: the generalized Fung form (including fully anisotropic and orthotropic cases) and the form proposed by Holzapfel, Gasser, and Ogden for arterial walls. Both forms are adequate for modeling soft biological tissue. However, whereas Fung's form is purely phenomenological, the Holzapfel-Gasser-Ogden form is micromechanically based.

In addition, Abaqus provides a general capability to support user-defined forms of the strain energy potential via two sets of user subroutines: one for strain-based and one for invariant-based formulations.

Generalized Fung form

The generalized Fung strain energy potential has the following form:

$$U = \frac{c}{2}(\exp(Q) - 1) + \frac{1}{D}\left(\frac{(J^{e\ell})^2 - 1}{2} - \ln J^{e\ell}\right),$$

where U is the strain energy per unit of reference volume; c and D are temperature-dependent material parameters; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion"; and Q is defined as

$$Q = \overline{\boldsymbol{\varepsilon}}^G : \mathbf{b} : \overline{\boldsymbol{\varepsilon}}^G = \overline{\varepsilon}^G_{ij} b_{ijkl} \overline{\varepsilon}^G_{kl},$$

where b_{ijkl} is a dimensionless symmetric fourth-order tensor of anisotropic material constants that can be temperature dependent and $\overline{\varepsilon}^G_{ij}$ are the components of the modified Green strain tensor.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = c\mathbf{b}, \qquad K_0 = \frac{2}{D}.$$

Abaqus supports two forms of the generalized Fung model: fully anisotropic and orthotropic. The number of independent components b_{ijlk} that must be specified depends on the level of anisotropy of the material: 21 for the fully anisotropic case and 9 for the orthotropic case.

Input File Usage: Use one of the following options:

*ANISOTROPIC HYPERELASTIC, FUNG-ANISOTROPIC
*ANISOTROPIC HYPERELASTIC, FUNG-ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**; **Material type: Anisotropic**; **Strain energy potential: Fung-Anisotropic** or **Fung-Orthotropic**

Holzapfel-Gasser-Ogden form

The form of the strain energy potential is based on that proposed by Holzapfel, Gasser, and Ogden (2000) and Gasser, Ogden, and Holzapfel (2006) for modeling arterial layers with distributed collagen fiber orientations:

$$U = C_{10}(\overline{I}_1 - 3) + \frac{1}{D}\left(\frac{(J^{e\ell})^2 - 1}{2} - \ln J^{e\ell}\right) + \frac{k_1}{2k_2}\sum_{\alpha=1}^{N}\left\{\exp[k_2\langle\overline{E}_\alpha\rangle^2] - 1\right\},$$

with

$$\overline{E}_\alpha \stackrel{\text{def}}{=} \kappa(\overline{I}_1 - 3) + (1 - 3\kappa)(\overline{I}_{4(\alpha\alpha)} - 1),$$

where U is the strain energy per unit of reference volume; C_{10}, D, k_1, k_2, and κ are temperature-dependent material parameters; N is the number of families of fibers ($N \leq 3$); $\overline{I}_1$ is the first deviatoric strain invariant; $J^{e\ell}$ is the elastic volume ratio as defined below in "Thermal expansion" and $\overline{I}_{4(\alpha\alpha)}$ are *pseudo-invariants* of $\overline{\mathbf{C}}$ and $\mathbf{A}_\alpha$.

The model assumes that the directions of the collagen fibers within each family are dispersed (with rotational symmetry) about a mean preferred direction. The parameter κ ($0 \leq \kappa \leq 1/3$) describes the level of dispersion in the fiber directions. If $\rho(\Theta)$ is the orientation density function that characterizes the distribution (it represents the normalized number of fibers with orientations in the range $[\Theta, \Theta + d\Theta]$ with respect to the mean direction), the parameter κ is defined as

$$\kappa = \frac{1}{4}\int_0^\pi \rho(\Theta)\sin^3\Theta d\Theta.$$

It is also assumed that all families of fibers have the same mechanical properties and the same dispersion. When $\kappa = 0$, the fibers are perfectly aligned (no dispersion). When $\kappa = 1/3$, the fibers are randomly distributed and the material becomes isotropic; this corresponds to a spherical orientation density function.

The strain-like quantity $\overline{E}_\alpha$ characterizes the deformation of the family of fibers with mean direction $\mathbf{A}_\alpha$. For perfectly aligned fibers ($\kappa = 0$), $\overline{E}_\alpha = \overline{I}_{4(\alpha\alpha)} - 1$; and for randomly distributed fibers ($\kappa = 1/3$), $\overline{E}_\alpha = (\overline{I}_1 - 3)/3$.

The first two terms in the expression of the strain energy function represent the distortional and volumetric contributions of the non-collagenous isotropic ground material, and the third term represents the contributions from the different families of collagen fibers, taking into account the effects of dispersion. A basic assumption of the model is that collagen fibers can support tension only, because they would buckle under compressive loading. Thus, the anisotropic contribution in the strain energy function appears only when the strain of the fibers is positive or, equivalently, when $\overline{E}_\alpha > 0$. This condition is enforced by the term $\langle \overline{E}_\alpha \rangle$, where the operator $\langle \cdot \rangle$ stands for the Macauley bracket and is defined as $\langle x \rangle = \frac{1}{2}(|x| + x)$.

See "Anisotropic hyperelastic modeling of arterial layers," Section 3.1.7 of the Abaqus Benchmarks Manual, for an example of an application of the Holzapfel-Gasser-Ogden energy potential to model arterial layers with distributed collagen fiber orientation.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = 4C_{10}\Im + 2(1-3\kappa)^2 k_1 \sum_{\alpha=1}^{N} H(\overline{E}_\alpha)\mathbf{A}_\alpha\mathbf{A}_\alpha\mathbf{A}_\alpha\mathbf{A}_\alpha, \qquad K_0 = \frac{2}{D},$$

where $\Im$ is the fourth-order unit tensor, and $H(x)$ is the Heaviside unit step function.

Input File Usage: *ANISOTROPIC HYPERELASTIC, HOLZAPFEL,
LOCAL DIRECTIONS=*N*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**; **Material type: Anisotropic**; **Strain energy potential: Holzapfel**, **Number of local directions:** *N*

User-defined form: strain-based

Alternatively, you can define the form of a strain-based strain energy potential directly with user subroutine **`UANISOHYPER_STRAIN`** in Abaqus/Standard or **`VUANISOHYPER_STRAIN`** in Abaqus/Explicit. The derivatives of the strain energy potential with respect to the components of the modified Green strain and the elastic volume ratio, $J^{e\ell}$, must be provided directly through these user subroutines.

Either compressible or incompressible behavior can be specified in Abaqus/Standard; only nearly incompressible behavior is allowed in Abaqus/Explicit.

Optionally, you can specify the number of property values needed as data in the user subroutine as well as the number of solution-dependent variables (see "User subroutines: overview," Section 18.1.1).

Input File Usage: In Abaqus/Standard use the following option to define compressible behavior:

*ANISOTROPIC HYPERELASTIC, USER, FORMULATION=STRAIN,
TYPE=COMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Standard use the following option to define incompressible behavior:

*ANISOTROPIC HYPERELASTIC, USER, FORMULATION=STRAIN, TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Explicit use the following option to define nearly incompressible behavior:

*ANISOTROPIC HYPERELASTIC, USER, FORMULATION=STRAIN, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**; **Material type: Anisotropic**, **Strain energy potential: User**, **Formulation: Strain**, **Type: Incompressible** or **Compressible**, **Number of property values:** *n*

User-defined form: invariant-based

Alternatively, you can define the form of an invariant-based strain energy potential directly with user subroutine **UANISOHYPER_INV** in Abaqus/Standard or **VUANISOHYPER_INV** in Abaqus/Explicit. Either compressible or incompressible behavior can be specified in Abaqus/Standard; only nearly incompressible behavior is allowed in Abaqus/Explicit.

Optionally, you can specify the number of property values needed as data in the user subroutine and the number of solution-dependent variables (see "User subroutines: overview," Section 18.1.1).

The derivatives of the strain energy potential with respect to the strain invariants must be provided directly through user subroutine **UANISOHYPER_INV** in Abaqus/Standard and **VUANISOHYPER_INV** in Abaqus/Explicit.

Input File Usage: In Abaqus/Standard use the following option to define compressible behavior:

*ANISOTROPIC HYPERELASTIC, USER, FORMULATION=INVARIANT, LOCAL DIRECTIONS=*N*, TYPE=COMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Standard use the following option to define incompressible behavior:

*ANISOTROPIC HYPERELASTIC, USER, FORMULATION=INVARIANT, LOCAL DIRECTIONS=*N*, TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Explicit use the following option to define nearly incompressible behavior:

*ANISOTROPIC HYPERELASTIC, USER, FORMULATION=INVARIANT, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**; **Material type: Anisotropic**, **Strain energy potential: User**, **Formulation: Invariant**, **Type: Incompressible** or **Compressible**, **Number of local directions:** *N*, **Number of property values:** *n*

Definition of preferred material directions

You must define the preferred material directions (or fiber directions) of the anisotropic hyperelastic material.

For strain-based forms (such as the Fung form and user-defined forms using user subroutines **UANISOHYPER_STRAIN** or **VUANISOHYPER_STRAIN**), you must specify a local orientation system ("Orientations," Section 2.2.5) to define the directions of anisotropy. Components of the modified Green strain tensor are calculated with respect to this system.

For invariant-based forms of the strain energy function (such as the Holzapfel form and user-defined forms using user subroutines **UANISOHYPER_INV** or **VUANISOHYPER_INV**), you must specify the local direction vectors, $\mathbf{A}_\alpha$, that characterize each family of fibers. These vectors need not be orthogonal in the initial configuration. Up to three local directions can be specified as part of the definition of a local orientation system ("Defining a local coordinate system directly" in "Orientations," Section 2.2.5); the local directions are referred to this system.

In Abaqus/CAE, the local direction vectors of the material are orthogonal and align with the axes of the assigned material orientation. The best practice is to assign the orientation using discrete orientations in Abaqus/CAE.

Material directions can be output to the output database as described in "Output," below.

Compressibility

Most soft tissues and fiber-reinforced elastomers have very little compressibility compared to their shear flexibility. This behavior does not warrant special attention for plane stress, shell, or membrane elements, but the numerical solution can be quite sensitive to the degree of compressibility for three-dimensional solid, plane strain, and axisymmetric elements. In cases where the material is highly confined (such as an O-ring used as a seal), the compressibility must be modeled correctly to obtain accurate results. In applications where the material is not highly confined, the degree of compressibility is typically not crucial; for example, it would be quite satisfactory in Abaqus/Standard to assume that the material is fully incompressible: the volume of the material cannot change except for thermal expansion.

Compressibility in Abaqus/Standard

In Abaqus/Standard the use of "hybrid" (mixed formulation) elements is required for incompressible materials. In plane stress, shell, and membrane elements the material is free to deform in the thickness direction. In this case special treatment of the volumetric behavior is not necessary; the use of regular stress/displacement elements is satisfactory.

Compressibility in Abaqus/Explicit

With the exception of the plane stress case, it is not possible to assume that the material is fully incompressible in Abaqus/Explicit because the program has no mechanism for imposing such a constraint at each material calculation point. Instead, some compressibility must be modeled. The difficulty is that, in many cases, the actual material behavior provides too little compressibility for the algorithms to work efficiently. Thus, except for the plane stress case, you must provide enough

compressibility for the code to work, knowing that this makes the bulk behavior of the model softer than that of the actual material. Failing to provide enough compressibility may introduce high frequency noise into the dynamic solution and require the use of excessively small time increments. Some judgment is, therefore, required to decide whether or not the solution is sufficiently accurate or whether the problem can be modeled at all with Abaqus/Explicit because of this numerical limitation.

If no value is given for the material compressibility of the anisotropic hyperelastic model, by default Abaqus/Explicit assumes the value $K_0/\mu_0 = 20$, where μ_0 is the largest value of the initial shear modulus (among the different material directions). The exception is for the case of user-defined forms, where some compressibility must be defined directly within user subroutine **UANISOHYPER_INV** or **VUANISOHYPER_INV**.

Thermal expansion

Both isotropic and orthotropic thermal expansion is permitted with the anisotropic hyperelastic material model.

The elastic volume ratio, $J^{e\ell}$, relates the total volume ratio, J, and the thermal volume ratio, J^{th}:

$$J^{e\ell} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = (1 + \varepsilon_1^{th})(1 + \varepsilon_2^{th})(1 + \varepsilon_3^{th}),$$

where ε_i^{th} are the principal thermal expansion strains that are obtained from the temperature and the thermal expansion coefficients ("Thermal expansion," Section 26.1.2).

Viscoelasticity

Anisotropic hyperelastic models can be used in combination with isotropic viscoelasticity to model rate-dependent material behavior ("Time domain viscoelasticity," Section 22.7.1). Because of the isotropy of viscoelasticity, the relaxation function is independent of the loading direction. This assumption may not be acceptable for modeling materials that exhibit strong anisotropy in their rate-dependent behavior; therefore, this option should be used with caution.

The anisotropic hyperelastic response of rate-dependent materials ("Time domain viscoelasticity," Section 22.7.1) can be specified by defining either the instantaneous response or the long-term response of such materials.

Input File Usage: Use either of the following options:

*ANISOTROPIC HYPERELASTIC, MODULI=INSTANTANEOUS
*ANISOTROPIC HYPERELASTIC, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**; **Material type: Anisotropic**; **Moduli: Long term** or **Instantaneous**

Stress softening

The response of typical anisotropic hyperelastic materials, such as reinforced rubbers and biological tissues, under cyclic loading and unloading usually displays stress softening effects during the first few cycles. After a few cycles the response of the material tends to stabilize and the material is said to be *pre-conditioned*. Stress softening effects, often referred to in the elastomers literature as Mullins effect, can be accounted for by using the anisotropic hyperelastic model in combination with the *pseudo-elasticity* model for Mullins effect in Abaqus (see "Mullins effect," Section 22.6.1). The stress softening effects provided by this model are isotropic.

Elements

The anisotropic hyperelastic material model can be used with solid (continuum) elements, finite-strain shells (except S4), continuum shells, and membranes. When used in combination with elements with plane stress formulations, Abaqus assumes fully incompressible behavior and ignores any amount of compressibility specified for the material.

Pure displacement formulation versus hybrid formulation in Abaqus/Standard

For continuum elements in Abaqus/Standard anisotropic hyperelasticity can be used with the pure displacement formulation elements or with the "hybrid" (mixed formulation) elements. Pure displacement formulation elements must be used with compressible materials, and "hybrid" (mixed formulation) elements must be used with incompressible materials.

In general, an analysis using a single hybrid element will be only slightly more computationally expensive than an analysis using a regular displacement-based element. However, when the wavefront is optimized, the Lagrange multipliers may not be ordered independently of the regular degrees of freedom associated with the element. Thus, the wavefront of a very large mesh of second-order hybrid tetrahedra may be noticeably larger than that of an equivalent mesh using regular second-order tetrahedra. This may lead to significantly higher CPU costs, disk space, and memory requirements.

Incompatible mode elements in Abaqus/Standard

Incompatible mode elements should be used with caution in applications involving large strains. Convergence may be slow, and in hyperelastic applications inaccuracies may accumulate. Erroneous stresses may sometimes appear in incompatible mode anisotropic hyperelastic elements that are unloaded after having been subjected to a complex deformation history.

Procedures

Anisotropic hyperelasticity must always be used with geometrically nonlinear analyses ("General and linear perturbation procedures," Section 6.1.3).

Output

In addition to the standard output identifiers available in Abaqus (“Abaqus/Standard output variable identifiers,” Section 4.2.1, and “Abaqus/Explicit output variable identifiers,” Section 4.2.2), local material directions will be output whenever element field output is requested to the output database. The local directions are output as field variables (LOCALDIR1, LOCALDIR2, LOCALDIR3) representing the direction cosines; these variables can be visualized as vector plots in the Visualization module of Abaqus/CAE (Abaqus/Viewer).

Output of local material directions is suppressed if no element field output is requested or if you specify not to have element material directions written to the output database (see “Specifying the directions for element output in Abaqus/Standard and Abaqus/Explicit” in “Output to the output database,” Section 4.1.3).

Additional references

- Gasser, T. C., R. W. Ogden, and G. A. Holzapfel, “Hyperelastic Modelling of Arterial Layers with Distributed Collagen Fibre Orientations,” Journal of the Royal Society Interface, vol. 3, pp. 15–35, 2006.
- Holzapfel, G. A., T. C. Gasser, and R. W. Ogden, “A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models,” Journal of Elasticity, vol. 61, pp. 1–48, 2000.
- Spencer, A. J. M., “Constitutive Theory for Strongly Anisotropic Solids,” A. J. M. Spencer (ed.), Continuum Theory of the Mechanics of Fibre-Reinforced Composites, CISM Courses and Lectures No. 282, International Centre for Mechanical Sciences, Springer-Verlag, Wien, pp. 1–32, 1984.

22.6 Stress softening in elastomers

- "Mullins effect," Section 22.6.1
- "Energy dissipation in elastomeric foams," Section 22.6.2

22.6.1 MULLINS EFFECT

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Combining material behaviors," Section 21.1.3
- "Elastic behavior: overview," Section 22.1.1
- "Hyperelastic behavior of rubberlike materials," Section 22.5.1
- "Anisotropic hyperelastic behavior," Section 22.5.3
- "Permanent set in rubberlike materials," Section 23.7.1
- "Energy dissipation in elastomeric foams," Section 22.6.2
- *HYPERELASTIC
- *MULLINS EFFECT
- *PLASTIC
- *UNIAXIAL TEST DATA
- *BIAXIAL TEST DATA
- *PLANAR TEST DATA
- "Mullins effect" in "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The Mullins effect model:

- is intended for modeling stress softening of filled rubber elastomers under quasi-static cyclic loading, a phenomenon referred to in the literature as Mullins effect;
- provides an extension to the well-known isotropic hyperelastic models;
- is based on the theory of incompressible isotropic elasticity modified by the addition of a single variable, referred to as the damage variable;
- assumes that only the deviatoric part of the material response is associated with damage;
- is intended for modeling material response in situations where different parts of the model undergo different levels of damage resulting in a different material response;
- is applied to the long-term modulus when combined with viscoelasticity; and
- cannot be used with hysteresis.

Abaqus provides a similar capability that can be applied to elastomeric foams (see "Energy dissipation in elastomeric foams," Section 22.6.2).

Material behavior

The real behavior of filled rubber elastomers under cyclic loading conditions is quite complex. Certain idealizations have been made for modeling purposes. In essence, these idealizations result in two main components to the material behavior: the first component describes the response of a material point (from an undeformed state) under monotonic straining, and the second component is associated with damage and describes the unloading-reloading behavior. The idealized response and the two components are described in the following sections.

Idealized material behavior

When an elastomeric test specimen is subjected to simple tension from its virgin state, unloaded, and then reloaded, the stress required on reloading is less than that on the initial loading for stretches up to the maximum stretch achieved during the initial loading. This stress softening phenomenon is known as the Mullins effect and reflects damage incurred during previous loading. This type of material response is depicted qualitatively in Figure 22.6.1–1.

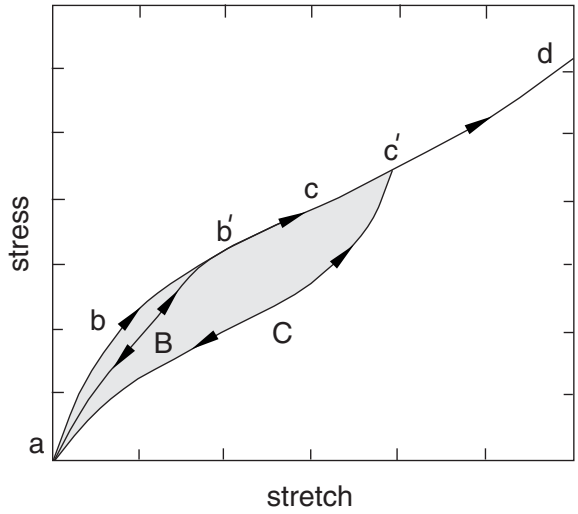


Figure 22.6.1–1 Idealized response of the Mullins effect model.

This figure and the accompanying description is based on work by Ogden and Roxburgh (1999), which forms the basis of the model implemented in Abaqus. Consider the primary loading path abb' of a previously unstressed material, with loading to an arbitrary point b'. On unloading from b', the path $b'Ba$ is followed. When the material is loaded again, the softened path is retraced as aBb'. If further loading is then applied, the path $b'c$ is followed, where $b'c$ is a continuation of the primary loading path $abb'cc'd$ (which is the path that would be followed if there was no unloading). If loading is now stopped at c', the path $c'Ca$ is followed on unloading and then retraced back to c' on reloading. If no further

loading beyond c' is applied, the curve aCc' represents the subsequent material response, which is then elastic. For loading beyond c', the primary path is again followed and the pattern described is repeated.

This is an ideal representation of Mullins effect since in practice there is some permanent set upon unloading and/or viscoelastic effects such as hysteresis. Points such as b' and c' may not exist in reality in the sense that unloading from the primary curve followed by reloading to the maximum strain level attained earlier usually results in a stress that is somewhat lower than the stress corresponding to the primary curve. In addition, the cyclic response for some filled elastomers shows evidence of progressive damage during unloading from and subsequent reloading to a certain maximum strain level. Such progressive damage usually occurs during the first few cycles, and the material behavior soon stabilizes to a loading/unloading cycle that is followed beyond the first few cycles. More details regarding the actual behavior and how test data that display such behavior can be used to calibrate the Abaqus model for Mullins effect are discussed later and in "Analysis of a solid disc with Mullins effect and permanent set," Section 3.1.7 of the Abaqus Example Problems Manual.

The loading path $abb'cc'd$ will henceforth be referred to as the "primary hyperelastic behavior." The primary hyperelastic behavior is defined by using a hyperelastic material model.

Stress softening is interpreted as being due to damage at the microscopic level. As the material is loaded, the damage occurs by the severing of bonds between filler particles and the rubber molecular chains. Different chain links break at different deformation levels, thereby leading to continuous damage with macroscopic deformation. An equivalent interpretation is that the energy required to cause the damage is not recoverable.

Primary hyperelastic behavior

Hyperelastic materials are described in terms of a "strain energy potential" function $U(\mathbf{F})$ that defines the strain energy stored in the material per unit reference volume (volume in the initial configuration). The quantity $\mathbf{F}$ is the deformation gradient tensor. To account for Mullins effect, Ogden and Roxburgh propose a material description that is based on an energy function of the form $U(\mathbf{F}, \eta)$, where the additional scalar variable, η, represents damage in the material. The damage variable controls the material properties in the sense that it enables the material response to be governed by an energy function on unloading and subsequent submaximal reloading different from that on the primary (initial) loading path from a virgin state. Because of the above interpretation of η, it is no longer appropriate to think of U as the stored elastic energy potential. Part of the energy is stored as strain energy, while the rest is dissipated due to damage. The shaded area in Figure 22.6.1–1 represents the energy dissipated by damage as a result of deformation until the point c', while the unshaded part represents the recoverable strain energy.

The following paragraphs provide a summary of the Mullins effect model in Abaqus. For further details, see "Mullins effect," Section 4.7.1 of the Abaqus Theory Manual. In preparation for writing the constitutive equations for Mullins effect, it is useful to separate the deviatoric and the volumetric parts of the total strain energy density as

$$U = U_{dev} + U_{vol}.$$

In the above equation U, U_{dev}, and U_{vol} are the total, deviatoric, and volumetric parts of the strain energy density, respectively. All the hyperelasticity models in Abaqus use strain energy potential functions that are already separated into deviatoric and volumetric parts. For example, the polynomial models use a strain energy potential of the form

$$U = \sum_{i+j=1}^{N} C_{ij}(\overline{I}_1 - 3)^i(\overline{I}_2 - 3)^j + \sum_{i=1}^{N} \frac{1}{D_i}(J^{e\ell} - 1)^{2i},$$

where the symbols have the usual interpretations. The first term on the right represents the deviatoric part of the elastic strain energy density function, and the second term represents the volumetric part.

Modified strain energy density function

The Mullins effect is accounted for by using an augmented energy function of the form

$$U(\overline{\lambda}_i, \eta) = \eta \tilde{U}_{dev}(\overline{\lambda}_i) + \phi(\eta) + \tilde{U}_{vol}(J^{e\ell}),$$

where $\tilde{U}_{dev}(\overline{\lambda}_i)$ is the deviatoric part of the strain energy density of the primary hyperelastic behavior, defined, for example, by the first term on the right-hand-side of the polynomial strain energy function given above; $\tilde{U}_{vol}(J^{e\ell})$ is the volumetric part of the strain energy density, defined, for example, by the second term on the right-hand-side of the polynomial strain energy function given above; $\overline{\lambda}_i$ $(i = 1, 2)$ represent the deviatoric principal stretches; and $J^{e\ell}$ represents the elastic volume ratio. The function $\phi(\eta)$ is a continuous function of the damage variable η and is referred to as the "damage function." When the deformation state of the material is on a point on the curve that represents the primary hyperelastic behavior, $\eta = 1$, $\phi(\eta) = 0$, $U(\overline{\lambda}_i, 1) = \tilde{U}_{dev}(\overline{\lambda}_i) + \tilde{U}_{vol}(J^{e\ell})$, and the augmented energy function reduces to the strain energy density function of the primary hyperelastic behavior. The damage variable varies continuously during the course of the deformation and always satisfies $0 < \eta \leq 1$. The above form of the energy function is an extension of the form proposed by Ogden and Roxburgh to account for material compressibility.

Stress computation

With the above modification to the energy function, the stresses are given by

$$\boldsymbol{\sigma}(\eta, \overline{\lambda}_i, J^{e\ell}) = \eta \tilde{\mathbf{S}}(\overline{\lambda}_i) - \tilde{p}(J^{e\ell})\mathbf{I},$$

where $\tilde{\mathbf{S}}$ is the deviatoric stress corresponding to the primary hyperelastic behavior at the current deviatoric deformation level $\overline{\lambda}_i$ and $\tilde{p}$ is the hydrostatic pressure of the primary hyperelastic behavior at the current volumetric deformation level $J^{e\ell}$. Thus, the deviatoric stress as a result of Mullins effect is obtained by simply scaling the deviatoric stress of the primary hyperelastic behavior with the damage variable η. The pressure stress is the same as that of the primary behavior. The model predicts loading/unloading along a single curve (that is different, in general, from the primary hyperelastic behavior) from any given strain level that passes through the origin of the stress-strain plot. It cannot

capture permanent strains upon removal of load. The model also predicts that a purely volumetric deformation will not have any damage or Mullins effect associated with it.

Damage variable

The damage variable, η, varies with the deformation according to

$$\eta = 1 - \frac{1}{r}\mathrm{erf}\left(\frac{U_{dev}^m - \tilde{U}_{dev}}{m + \beta U_{dev}^m}\right),$$

where U_{dev}^m is the maximum value of $\tilde{U}_{dev}$ at a material point during its deformation history; r, β, and m are material parameters; and $\mathrm{erf}(x)$ is the error function defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-w^2)dw.$$

When $\tilde{U}_{dev} = U_{dev}^m$, corresponding to a point on the primary curve, $\eta = 1.0$. On the other hand, η attains its minimum value, η_m, given by

$$\eta_m = 1 - \frac{1}{r}\mathrm{erf}\left(\frac{U_{dev}^m}{m + \beta U_{dev}^m}\right),$$

upon removal of deformation, when $\tilde{U}_{dev} = 0$. For all intermediate values of $\tilde{U}_{dev}$, η varies monotonically between 1.0 and η_m. While the parameters r and β are dimensionless, the parameter m has the dimensions of energy. The equation for η reduces to that proposed by Ogden and Roxburgh when $\beta = 0$. The material parameters may be specified directly or may be computed by Abaqus based on curve-fitting of unloading-reloading test data. These parameters are subject to the restrictions $r > 1$, $\beta \geq 0$, and $m \geq 0$ (the parameters β and m cannot both be zero). Alternatively, the damage variable η can be defined through user subroutine **UMULLINS** in Abaqus/Standard and **VUMULLINS** in Abaqus/Explicit.

If the parameter $\beta = 0$ and the parameter m has a value that is small compared to U_{dev}^m, the slope of the stress-strain curve at the initiation of unloading from relatively large strain levels may become very high. As a result, the response may become discontinuous, as illustrated in Figure 22.6.1–2. This kind of behavior may lead to convergence problems in Abaqus/Standard. In Abaqus/Explicit the high stiffness will lead to very small stable time increments, thereby leading to a degradation in performance. This problem can be avoided by choosing a small value for β. The choice $\beta = 0$ can be used to define the original Ogden-Roxburgh model. In Abaqus/Standard the default value of β is 0. In Abaqus/Explicit, however, the default value of β is 0.1. Thus, if you do not specify a value for β, it is assumed to be 0 in Abaqus/Standard and 0.1 in Abaqus/Explicit.

The parameters r, β, and m do not have direct physical interpretations in general. The parameter m controls whether damage occurs at low strain levels. If $m = 0$, there is a significant amount of damage at low strain levels. On the other hand, a nonzero m leads to little or no damage at low strain levels. For further discussion regarding the implications of this model to the energy dissipation, see “Mullins

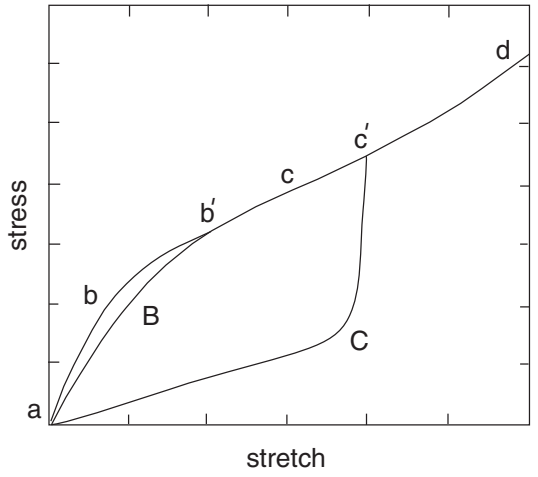


Figure 22.6.1–2 Overly stiff response at the initiation of unloading.

effect," Section 4.7.1 of the Abaqus Theory Manual. The qualitative effects of varying the parameters r and β individually, while holding the other parameters fixed, are shown in Figure 22.6.1–3.

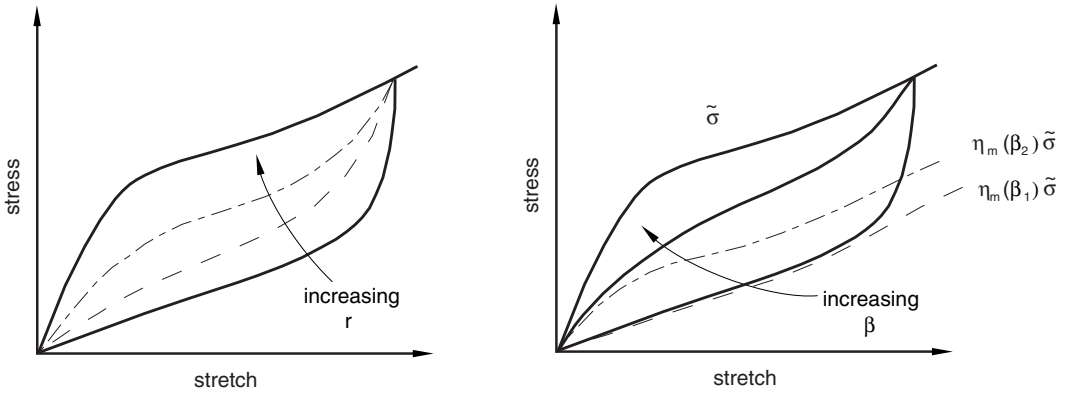


Figure 22.6.1–3 Qualitative dependence of damage on material properties.

The left figure shows the unloading-reloading curve from a certain maximum strain level for increasing values of r. It suggests that the parameter r controls the amount of damage, with decreasing damage for increasing r. This behavior follows from the fact that the larger the value of r, the less the damage variable η can deviate from unity. The figure on the right shows the unloading-reloading curve from a certain maximum strain level for increasing values of β. The figure suggests that increasing β also leads to lower amounts of damage. It also shows that the unloading-reloading response approaches the asymptotic response given by $\eta_m\tilde{\sigma}$, where η_m is the minimum value of η, faster for lower values of β.

The dashed curves represent the asymptotic response at two different values of β (β_1 and β_2). For fixed values of r and m, η_m is a function of β. In particular, if $m = 0$,

$$\eta_m = 1 - \frac{1}{r}\mathrm{erf}\left(\frac{1}{\beta}\right).$$

The above relation is approximately true if U_{dev}^m is much greater than m.

Specifying the Mullins effect material model in Abaqus

The primary hyperelastic behavior is defined by using the hyperelastic material model (see "Hyperelastic behavior of rubberlike materials," Section 22.5.1). The Mullins effect model can be defined by specifying the Mullins effect parameters directly or by using test data to calibrate the parameters. Alternatively, you can define the Mullins effect model with user subroutine **UMULLINS** in Abaqus/Standard and **VUMULLINS** in Abaqus/Explicit.

Specifying the parameters directly

The parameters r, m, and β of the Mullins effect can be given directly as functions of temperature and/or field variables.

Input File Usage: *MULLINS EFFECT

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Damage for Elastomers→Mullins Effect:
Definition: **Constants**

Using test data to calibrate the parameters

Experimental unloading-reloading data from different strain levels can be specified for up to three simple tests: uniaxial, biaxial, and planar. Abaqus will then compute the material parameters using a nonlinear least-squares curve fitting algorithm. It is generally best to obtain data from several experiments involving different kinds of deformation over the range of strains of interest in the actual application and to use all these data to determine the parameters. It is also important to obtain a good curve-fit for the primary hyperelastic behavior if the primary behavior is defined using test data.

By default, Abaqus attempts to fit all three parameters to the given data. This is possible in general, except in the situation when the test data correspond to unloading-reloading from only a single value of U_{dev}^m. In this case the parameters m and β cannot be determined independently; one of them must be specified. If you specify neither m nor β, Abaqus needs to assume a default value for one of these parameters. In light of the potential problems discussed earlier with $\beta = 0$, Abaqus assumes that $m = 0$ in the above situation. The curve-fitting may also be carried out by specifying any one or two of the material parameters to be fixed, predetermined values.

As many data points as required can be entered from each test. It is recommended that data from all three tests (on samples taken from the same piece of material) be included and that the data points cover unloading/reloading from/to the range of nominal strain expected to arise in the actual loading.

The strain data should be given as nominal strain values (change in length per unit of original length). The stress data should be given as nominal stress values (force per unit of original cross-sectional area).

These tests allow for entering both compression and tension data. Compressive stresses and strains are entered as negative values.

For each set of test input, the data point with the maximum nominal strain identifies the point of unloading. This point is used by the curve-fitting algorithm to compute U^m_{dev} for that curve.

Figure 22.6.1–4 shows some typical unloading-reloading data from three different strain levels.

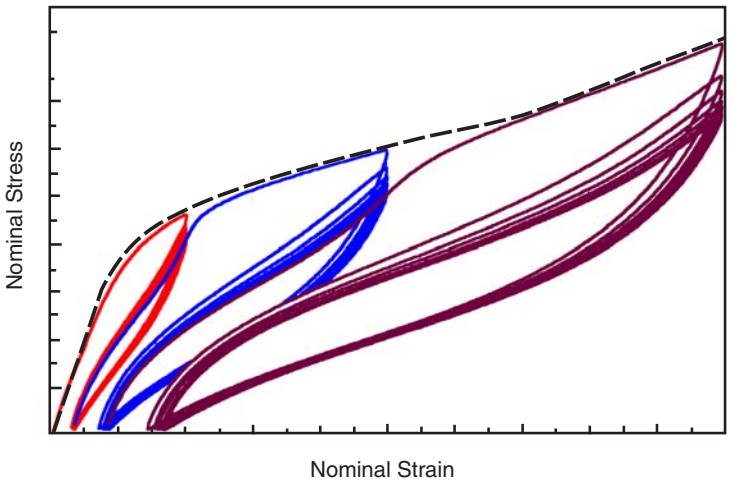


Figure 22.6.1–4 Typical available test data for Mullins effect.

The data include multiple loading and unloading cycles from each strain level. As Figure 22.6.1–4 indicates, the loading/unloading cycles from any given strain level do not occur along a single curve, and there is some amount of hysteresis. There is also some amount of permanent set upon removal of the applied load. The data also show evidence of progressive damage with repeated cycling at any given maximum strain level. The response appears to stabilize after a number of cycles. When such data are used to calibrate the Mullins effect model, the resulting response will capture the overall stiffness characteristics, while ignoring effects such as hysteresis, permanent set, or progressive damage. The above data can be provided to Abaqus in the following manner:

- The primary curve can be made up of the data points indicated by the dashed curve in Figure 22.6.1–4. Essentially, this consists of an envelope of the first loading curves to the different strain levels.
- The unloading-reloading curves from the three different strain levels can be specified by providing the data points as is; i.e., as the repeated unloading-reloading cycles shown in Figure 22.6.1–4. As discussed earlier, the data from the different strain levels need to be distinguished by providing them as different tables. For example, assuming that the test data correspond to the uniaxial tension state, three tables of uniaxial test data would have to be defined for the three different strain levels shown in Figure 22.6.1–4. In this case Abaqus will provide a best fit using all the data points (from all strain levels). The resulting fit would result in a response that is an average of all the test data

at any given strain level. While permanent set may be modeled (see "Permanent set in rubberlike materials," Section 23.7.1), hysteresis will be lost in the process.

- Alternatively, you may provide any one unloading-reloading cycle from each different strain level. If the component is expected to undergo repeated cyclic loading, the latter may be, for example, the stabilized cycle at each strain level. On the other hand, if the component is expected to undergo predominantly monotonic loading with perhaps small amounts of unloading, the very first unloading curve at each strain level may be the appropriate input data for calibrating the Mullins coefficients.

Once the Mullins effect constants are determined, the behavior of the Mullins effect model in Abaqus is established. However, the quality of this behavior must be assessed: the prediction of material behavior under different deformation modes must be compared against the experimental data. You must judge whether the Mullins effect constants determined by Abaqus are acceptable, based on the correlation between the Abaqus predictions and the experimental data. Single-element test cases can be used to derive the nominal stress–nominal strain response of the material model.

The steps that can be taken for improving the quality of the fit for the Mullins effect parameters are similar in essence to the guidelines provided for curve fitting the primary hyperelastic behavior (see "Hyperelastic behavior of rubberlike materials," Section 22.5.1, for details). In addition, the quality of the fit for the Mullins effect parameters depends on a good fit for the primary hyperelastic behavior, if the primary behavior is defined using test data.

The quality of the fit can be evaluated by carrying out a numerical experiment with a single element that is loaded in the same mode for which test data has been provided. Alternatively, the numerical response for both the primary and the softening behavior can be obtained by requesting model definition data output (see "Output," Section 4.1.1) and carrying out a data check analysis. The response computed by Abaqus is printed in the data (`.dat`) file along with the experimental data. This tabular data can be plotted in Abaqus/CAE for comparison and evaluation purposes. The primary hyperelastic behavior can also be evaluated with the automated material evaluation tools in Abaqus/CAE.

Input File Usage: ∗MULLINS EFFECT, TEST DATA INPUT, BETA *and/or* M *and/or* R

In addition, use at least one and up to three of the following options to give the unloading-reloading test data (see "Experimental tests" in the section describing hyperelastic test data input, "Hyperelastic behavior of rubberlike materials," Section 22.5.1):

∗UNIAXIAL TEST DATA
∗BIAXIAL TEST DATA
∗PLANAR TEST DATA

Multiple unloading-reloading curves from different strain levels for any given test type can be entered by repeated specification of the appropriate test data option.

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Damage for Elastomers→Mullins Effect: **Definition**: **Test Data Input**: enter the values for up to two of the values **r**, **m**, and

beta. In addition, select and enter data for at least one of the following:
Add Test→Biaxial Test, **Planar Test**, or **Uniaxial Test**

User subroutine specification

An alternative method for defining the Mullins effect involves defining the damage variable in user subroutine **UMULLINS** in Abaqus/Standard and **VUMULLINS** in Abaqus/Explicit. Optionally, you can specify the number of property values needed as data in the user subroutine. You must provide the damage variable, η, and its derivative, $\frac{d\eta}{d\bar{U}_{dev}}$. The latter contributes to the Jacobian of the overall system of equations and is necessary to ensure good convergence characteristics in Abaqus/Standard. If needed, you can specify the number of solution-dependent variables ("User subroutines: overview," Section 18.1.1). These solution-dependent variables can be updated in the user subroutine. The damage dissipation energy and the recoverable part of the energy may also be defined for output purposes.

User subroutines **UMULLINS** and **VUMULLINS** can be used in combination with all hyperelastic potentials in Abaqus, including user-defined potentials (via user subroutines **UHYPER**, **UANISOHYPER_INV**, and **UANISOHYPER_STRAIN** Abaqus/Standard, and **VUANISOHYPER_INV** and **VUANISOHYPER_STRAIN** in Abaqus/Explicit).

Input File Usage: *MULLINS EFFECT, USER, PROPERTIES=*constants*

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Damage for Elastomers→Mullins Effect:
Definition: **User Defined**

Viscoelasticity

When viscoelasticity is used in combination with Mullins effect, stress softening is applied to the long-term behavior.

In this case specification of the parameter m (which has units of energy) should be done carefully. If the underlying hyperelastic behavior is defined with an instantaneous modulus, m will be interpreted to be instantaneous. Otherwise, m is considered to be long term.

Elements

The Mullins effect material model can be used with all element types that support the use of the hyperelastic material model.

Procedures

The Mullins effect material model can be used in all procedure types that support the use of the hyperelastic material model. In linear perturbation steps in Abaqus/Standard the current material tangent stiffness is used to determine the response. Specifically, when a linear perturbation is carried out about a base state that is on the primary curve, the unloading tangent stiffness will be used.

In Abaqus/Explicit the unloading tangent stiffness is always used to compute the stable time increment. As a result, the inclusion of Mullins effect leads to more increments in the analysis, even when no unloading actually takes place.

The Mullins effect material model can also be used in a steady-state transport analysis in Abaqus/Standard to obtain steady-state rolling solutions. Issues related to the use of the Mullins effect in a steady-state transport analysis can be found in "Steady-state transport analysis," Section 6.4.1, and "Analysis of a solid disc with Mullins effect and permanent set," Section 3.1.7 of the Abaqus Example Problems Manual.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning for the Mullins effect material model:

DMENER	Energy dissipated per unit volume by damage.
ELDMD	Total energy dissipated in element by damage.
ALLDMD	Energy dissipated in whole (or partial) model by damage. The contribution from ALLDMD is included in the total strain energy ALLIE.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
SENER	The recoverable part of the energy per unit volume.
ELSE	The recoverable part of the energy in the element.
ALLSE	The recoverable part of the energy in the whole (partial) model.
ESEDEN	The recoverable part of the energy per unit volume in the element.

The damage energy dissipation, represented by the shaded area in Figure 22.6.1–1 for deformation until $\bar{c}$, is computed as follows. When the damaged material is in a fully unloaded state, the augmented energy function has the residual value $U(\mathbf{I}, \eta_m) = \phi(\eta_m)$. The residual value of the energy function upon complete unloading represents the energy dissipated due to damage in the material. The recoverable part of the energy is obtained by subtracting the dissipated energy from the augmented energy as $\eta\tilde{U}_{dev}(\overline{\lambda}_i) + \phi(\eta) + \tilde{U}_{vol} - \phi(\eta_m)$.

The damage energy accumulates with progressive deformation along the primary curve and remains constant during unloading. During unloading, the recoverable part of the strain energy is released. The latter becomes zero when the material point is completely unloaded. Upon further reloading from a completely unloaded state, the recoverable part of the strain energy increases from zero. When the maximum strain that was attained earlier is exceeded upon reloading, further accumulation of damage energy occurs.

Additional reference

- Ogden, R. W., and D. G. Roxburgh, "A Pseudo-Elastic Model for the Mullins Effect in Filled Rubber," Proceedings of the Royal Society of London, Series A, vol. 455, p. 2861–2877, 1999.

22.6.2 ENERGY DISSIPATION IN ELASTOMERIC FOAMS

Products: Abaqus/Standard Abaqus/Explicit

References

- "Material library: overview," Section 21.1.1
- "Combining material behaviors," Section 21.1.3
- "Elastic behavior: overview," Section 22.1.1
- "Hyperelastic behavior in elastomeric foams," Section 22.5.2
- "Mullins effect," Section 22.6.1
- *HYPERFOAM
- *MULLINS EFFECT
- *UNIAXIAL TEST DATA
- *BIAXIAL TEST DATA
- *PLANAR TEST DATA

Overview

Energy dissipation in elastomeric foams in Abaqus:

- allows the modeling of permanent energy dissipation and stress softening effects in elastomeric foams;
- uses an approach based on the Mullins effect for elastomeric rubbers ("Mullins effect," Section 22.6.1);
- provides an extension to the isotropic elastomeric foam model ("Hyperelastic behavior in elastomeric foams," Section 22.5.2);
- is intended for modeling energy absorption in foam components subjected to dynamic loading under deformation rates that are high compared to the characteristic relaxation time of the foam; and
- cannot be used with viscoelasticity.

Energy dissipation in elastomeric foams

Abaqus provides a mechanism to include permanent energy dissipation and stress softening effects in elastomeric foams. The approach is similar to that used to model the Mullins effect in elastomeric rubbers, described in "Mullins effect," Section 22.6.1. The functionality is primarily intended for modeling energy absorption in foam components subjected to dynamic loading under deformation rates that are high compared to the characteristic relaxation time of the foam; in such cases it is acceptable to assume that the foam material is damaged permanently.

The material response is depicted qualitatively in Figure 22.6.2–1.

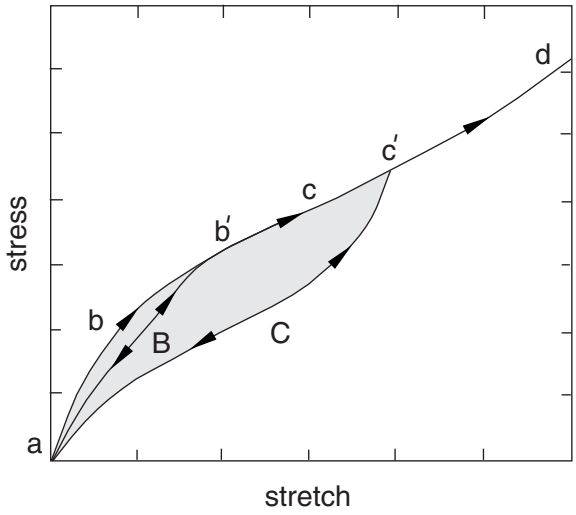


Figure 22.6.2–1 Typical stress-stretch response of an elastomeric foam material with energy dissipation.

Consider the primary loading path $abb^{'}$ of a previously unstressed foam, with loading to an arbitrary point $b^{'}$. On unloading from $b^{'}$, the path $b^{'}Ba$ is followed. When the material is loaded again, the softened path is retraced as $aBb^{'}$. If further loading is then applied, the path $b^{'}c$ is followed, where $b^{'}c$ is a continuation of the primary loading path $abb^{'}cc^{'}d$ (which is the path that would be followed if there were no unloading). If loading is now stopped at $c^{'}$, the path $c^{'}Ca$ is followed on unloading and then retraced back to $c^{'}$ on reloading. If no further loading beyond $c^{'}$ is applied, the curve $aCc^{'}$ represents the subsequent material response, which is then elastic. For loading beyond $c^{'}$, the primary path is again followed and the pattern described is repeated. The shaded area in Figure 22.6.2–1 represents the energy dissipated by damage in the material for deformation until $c^{'}$.

Modified strain energy density function

Energy dissipation effects are accounted for by introducing an augmented strain energy density function of the form

$$U(\hat{\lambda}_i, \eta) = \eta \tilde{U}(\hat{\lambda}_i) + \phi(\eta),$$

where $\hat{\lambda}_i$ $(i = 1, 2, 3)$ represent the principal mechanical stretches and $\tilde{U}(\hat{\lambda}_i)$ is the strain energy potential for the primary foam behavior described in "Hyperelastic behavior in elastomeric foams," Section 22.5.2, defined by the polynomial strain energy function

$$\tilde{U}(\hat{\lambda}_i) = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left[\hat{\lambda}_1^{\alpha_i} + \hat{\lambda}_2^{\alpha_i} + \hat{\lambda}_3^{\alpha_i} - 3 + \frac{1}{\beta_i} \left((J^{e\ell})^{-\alpha_i \beta_i} - 1 \right) \right].$$

The function $\phi(\eta)$ is a continuous function of the damage variable, η, and is referred to as the "damage function." The damage variable varies continuously during the course of the deformation and always satisfies $0 < \eta \leq 1$, with $\eta = 1$ on the points of the primary curve. The damage function $\phi(\eta)$ satisfies the condition $\phi(1) = 0$; thus, when the deformation state of the material is on a point on the curve that represents the primary foam behavior, $U(\hat{\lambda}_i, 1) = \tilde{U}(\hat{\lambda}_i)$ and the augmented energy function reduces to the strain energy potential for the primary foam behavior.

The above expression of the augmented strain energy density function is similar to the form proposed by Ogden and Roxburgh to model the Mullins effect in filled rubber elastomers (see "Mullins effect," Section 22.6.1), with the difference that in the case of elastomeric foams an augmentation of the total strain energy (including the volumetric part) is considered. This modification is required for the model to predict energy absorption under pure hydrostatic loading of the foam.

Stress computation

With the above modification to the energy function, the stresses are given by

$$\boldsymbol{\sigma}(\eta, \hat{\lambda}_i) = \eta \tilde{\boldsymbol{\sigma}}(\hat{\lambda}_i),$$

where $\tilde{\boldsymbol{\sigma}}$ is the stress corresponding to the primary foam behavior at the current deformation level $\hat{\lambda}_i$. Thus, the stress is obtained by simply scaling the stress of the primary foam behavior by the damage variable, η. From any given strain level the model predicts unloading/reloading along a single curve (that is different, in general, from the primary foam behavior) that passes through the origin of the stress-strain plot. The model also predicts energy dissipation under purely volumetric deformation.

Damage variable

The damage variable, η, varies with the deformation according to

$$\eta = 1 - \frac{1}{r} \mathrm{erf} \left(\frac{U^m - \tilde{U}}{m + \beta U^m} \right),$$

where U^m is the maximum value of $\tilde{U}$ at a material point during its deformation history; r, β, and m are material parameters; and $\mathrm{erf}(x)$ is the error function. When $\tilde{U} = U^m$, corresponding to a point on the primary curve, $\eta = 1.0$. On the other hand, upon removal of deformation, when $\tilde{U} = 0$, the damage variable, η, attains its minimum value, η_m, given by

$$\eta_m = 1 - \frac{1}{r} \mathrm{erf} \left(\frac{U^m}{m + \beta U^m} \right).$$

For all intermediate values of $\tilde{U}$, η varies monotonically between 1.0 and η_m. While the parameters r and β are dimensionless, the parameter m has the dimensions of energy. The material parameters can

be specified directly or can be computed by Abaqus based on curve fitting of unloading-reloading test data. These parameters are subject to the restrictions $r > 1$, $\beta \geq 0$, and $m \geq 0$ (the parameters β and m cannot both be zero). Alternatively, the damage variable η can be defined through user subroutine **UMULLINS** in Abaqus/Standard and **VUMULLINS** in Abaqus/Explicit.

If the parameter $\beta = 0$ and the parameter m has a value that is small compared to U^m, the slope of the stress-strain curve at the initiation of unloading from relatively large strain levels may become very high. As a result, the response may become discontinuous. This kind of behavior may lead to convergence problems in Abaqus/Standard. In Abaqus/Explicit the high stiffness will lead to very small stable time increments, thereby leading to a degradation in performance. This problem can be avoided by choosing a small value for β. In Abaqus/Standard the default value of β is 0. In Abaqus/Explicit, however, the default value of β is 0.1. Thus, if you do not specify a value for β, it is assumed to be 0 in Abaqus/Standard and 0.1 in Abaqus/Explicit.

The parameters r, β, and m do not have direct physical interpretations in general. The parameter m controls whether damage occurs at low strain levels. If $m = 0$, there is a significant amount of damage at low strain levels. On the other hand, a nonzero m leads to little or no damage at low strain levels. For further discussion regarding the implications of this model on the energy dissipation, see "Mullins effect," Section 4.7.1 of the Abaqus Theory Manual.

Specifying properties for energy dissipation in elastomeric foams

The primary elastomeric foam behavior is defined by using the hyperfoam material model. Energy dissipation can be defined by specifying the parameters in the expression of the damage variable directly or by using test data to calibrate the parameters. Alternatively, you can define the Mullins effect model with user subroutine **UMULLINS** in Abaqus/Standard and **VUMULLINS** in Abaqus/Explicit.

Specifying the parameters directly

The parameters r, m, and β in the expression of the damage variable can be given directly as functions of temperature and/or field variables.

Input File Usage: *MULLINS EFFECT

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Damage for Elastomers→Mullins Effect:
Definition: **Constants**

Using test data to calibrate the parameters

Experimental unloading-reloading data from different strain levels can be specified for up to three simple tests: uniaxial, biaxial, and planar. Abaqus will then compute the material parameters using a nonlinear least-squares curve fitting algorithm. See "Mullins effect," Section 22.6.1, for a detailed discussion of this approach.

Input File Usage: *MULLINS EFFECT, TEST DATA INPUT, BETA *and/or* M *and/or* R

In addition, use at least one and up to three of the following options to give the unloading-reloading test data:

*UNIAXIAL TEST DATA

*BIAXIAL TEST DATA
*PLANAR TEST DATA

Multiple unloading-reloading curves from different strain levels for any given test type can be entered by repeated specification of the appropriate test data option.

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Damage for Elastomers→Mullins Effect: **Definition**: **Test Data Input**: enter the values for up to two of the values **r**, **m**, and **beta**. In addition, enter data for at least one of the following **Suboptions→Biaxial Test**, **Planar Test**, or **Uniaxial Test**

User subroutine specification

An alternative method for specifying energy dissipation involves defining the damage variable in user subroutine **UMULLINS** in Abaqus/Standard and **VUMULLINS** in Abaqus/Explicit. Optionally, you can specify the number of property values needed as data in the user subroutine. You must provide the damage variable, η, and its derivative, $\frac{d\eta}{dU}$. The latter contributes to the Jacobian of the overall system of equations and is necessary to ensure good convergence characteristics in Abaqus/Standard. If needed, you can specify the number of solution-dependent variables ("User subroutines: overview," Section 18.1.1). These solution-dependent variables can be updated in the user subroutine. The damage dissipation energy and the recoverable part of the energy can also be defined for output purposes.

Input File Usage: *MULLINS EFFECT, USER, PROPERTIES=*constants*

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Damage for Elastomers→Mullins Effect: **Definition**: **User Defined**

Elements

The model can be used with all element types that support the use of the elastomeric foam material model.

Procedures

The model can be used in all procedure types that support the use of the elastomeric foam material model. In linear perturbation steps in Abaqus/Standard the current material tangent stiffness is used to determine the response. Specifically, when a linear perturbation is carried out about a base state that is on the primary curve, the unloading tangent stiffness will be used.

In Abaqus/Explicit the unloading tangent stiffness is always used to compute the stable time increment. As a result, the inclusion of stress-softening effects may lead to more increments in the analysis, even when no unloading actually takes place.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning when energy dissipation is present in the model:

DMENER	Energy dissipated per unit volume by damage.
ELDMD	Total energy dissipated in element by damage.
ALLDMD	Energy dissipated in whole (or partial) model by damage. The contribution from ALLDMD is included in the total strain energy ALLIE.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
SENER	The recoverable part of the energy per unit volume.
ELSE	The recoverable part of the energy in the element.
ALLSE	The recoverable part of the energy in the whole (partial) model.
ESEDEN	The recoverable part of the energy per unit volume in the element.

The damage energy dissipation, represented by the shaded area in Figure 22.6.2–1 for deformation until $c^{'}$, is computed as follows. When the damaged material is in a fully unloaded state, the augmented energy function has the residual value $U(\mathbf{I}, \eta_m) = \phi(\eta_m)$. The residual value of the energy function upon complete unloading represents the energy dissipated due to damage in the material. The recoverable part of the energy is obtained by subtracting the dissipated energy from the augmented energy as $\eta\tilde{U}(\hat{\lambda}_i) + \phi(\eta) - \phi(\eta_m)$.

The damage energy accumulates with progressive deformation along the primary curve and remains constant during unloading. During unloading, the recoverable part of the strain energy is released. The latter becomes zero when the material point is unloaded completely. Upon further reloading from a completely unloaded state, the recoverable part of the strain energy increases from zero. When the maximum strain that was attained earlier is exceeded upon reloading, further accumulation of damage energy occurs.

22.7 Viscoelasticity

- “Time domain viscoelasticity,” Section 22.7.1
- “Frequency domain viscoelasticity,” Section 22.7.2

22.7.1 TIME DOMAIN VISCOELASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Frequency domain viscoelasticity," Section 22.7.2
- *VISCOELASTIC
- *SHEAR TEST DATA
- *VOLUMETRIC TEST DATA
- *COMBINED TEST DATA
- *TRS
- "Defining time domain viscoelasticity" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The time domain viscoelastic material model:

- describes isotropic rate-dependent material behavior for materials in which dissipative losses primarily caused by "viscous" (internal damping) effects must be modeled in the time domain;
- assumes that the shear (deviatoric) and volumetric behaviors are independent in multiaxial stress states (except when used for an elastomeric foam);
- can be used only in conjunction with "Linear elastic behavior," Section 22.2.1; "Hyperelastic behavior of rubberlike materials," Section 22.5.1; or "Hyperelastic behavior in elastomeric foams," Section 22.5.2, to define the continuum elastic material properties;
- can be used in Abaqus/Explicit with "Linear elastic traction-separation behavior" in "Defining the constitutive response of cohesive elements using a traction-separation description," Section 32.5.6;
- is active only during a transient static analysis ("Quasi-static analysis," Section 6.2.5), a transient implicit dynamic analysis ("Implicit dynamic analysis using direct integration," Section 6.3.2), an explicit dynamic analysis ("Explicit dynamic analysis," Section 6.3.3), a steady-state transport analysis ("Steady-state transport analysis," Section 6.4.1), a fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3), a fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4), or a transient (consolidation) coupled pore fluid diffusion and stress analysis ("Coupled pore fluid diffusion and stress analysis," Section 6.8.1);
- can be used in large-strain problems;

- can be calibrated using time-dependent creep test data, time-dependent relaxation test data, or frequency-dependent cyclic test data; and
- can be used to couple viscous dissipation with the temperature field in a fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3) or a fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4).

Defining the shear behavior

Time domain viscoelasticity is available in Abaqus for small-strain applications where the rate-independent elastic response can be defined with a linear elastic material model and for large-strain applications where the rate-independent elastic response must be defined with a hyperelastic or hyperfoam material model.

Small strain

Consider a shear test at small strain in which a time varying shear strain, $\gamma(t)$, is applied to the material. The response is the shear stress $\tau(t)$. The viscoelastic material model defines $\tau(t)$ as

$$\tau(t) = \int_0^t G_R(t-s)\dot{\gamma}(s)\,ds,$$

where $G_R(t)$ is the time-dependent "shear relaxation modulus" that characterizes the material's response. This constitutive behavior can be illustrated by considering a relaxation test in which a strain γ is suddenly applied to a specimen and then held constant for a long time. The beginning of the experiment, when the strain is suddenly applied, is taken as zero time, so that

$$\tau(t) = \int_0^t G_R(t-s)\dot{\gamma}(s)\,ds = G_R(t)\gamma \quad (\text{since } \dot{\gamma} = 0 \text{ for } t > 0),$$

where γ is the fixed strain. The viscoelastic material model is "long-term elastic" in the sense that, after having been subjected to a constant strain for a very long time, the response settles down to a constant stress; i.e., $G_R(t) \to G_\infty$ as $t \to \infty$.

The shear relaxation modulus can be written in dimensionless form:

$$g_R(t) = G_R(t)/G_0,$$

where $G_0 = G_R(0)$ is the instantaneous shear modulus, so that the expression for the stress takes the form

$$\tau(t) = G_0 \int_0^t g_R(t-s)\dot{\gamma}(s)\,ds.$$

The dimensionless relaxation function has the limiting values $g_R(0) = 1$ and $g_R(\infty) = G_\infty/G_0$.

Anisotropic elasticity in Abaqus/Explicit

The equation for the shear stress can be transformed by using integration by parts:

$$\tau(t) = G_0 \left(\gamma + \int_0^t \dot{g}_R(s)\gamma(t-s)\, ds \right).$$

It is convenient to write this equation in the form

$$\tau(t) = \tau_0(t) + \int_0^t \dot{g}_R(s)\tau_0(t-s)\, ds,$$

where $\tau_0(t)$ is the instantaneous shear stress at time t. This can be generalized to multi-dimensions as

$$\boldsymbol{\tau}(t) = \boldsymbol{\tau}_0(t) + \int_0^t \dot{g}_R(s)\boldsymbol{\tau}_0(t-s)\, ds,$$

where $\boldsymbol{\tau}(t)$ is the deviatoric part of the stress tensor and $\boldsymbol{\tau}_0(t)$ is the deviatoric part of the instantaneous stress tensor. Here the viscoelasticity is assumed to be isotropic; i.e., the relaxation function is independent of the loading direction.

This form allows a straightforward generalization to anisotropic elastic deformations, where the deviatoric part of instantaneous stress tensor is computed as $\boldsymbol{\tau}_0(t) = \overline{\mathbf{D}}_0 : \mathbf{e}$. Here $\overline{\mathbf{D}}_0$ is the instantaneous deviatoric elasticity tensor, and $\mathbf{e}$ is the deviatoric part of the strain tensor.

Large strain

The above form also allows a straightforward generalization to nonlinear elastic deformations, where the deviatoric part of the instantaneous stress $\boldsymbol{\tau}_0(t)$ is computed using a hyperelastic strain enery potential. This generalization yields a linear viscoelasticity model, in the sense that the dimensionless stress relaxation function is independent of the magnitude of the deformation.

In the above equation the instantaneous stress, $\boldsymbol{\tau}_0$, applied at time $t - s$ influences the stress, $\boldsymbol{\tau}$, at time t. Therefore, to create a proper finite-strain formulation, it is necessary to map the stress that existed in the configuration at time $t - s$ into the configuration at time t. In Abaqus this is done by means of the "standard-push-forward" transformation with the relative deformation gradient $\mathbf{F}_{t-s}(t)$:

$$\mathbf{F}_{t-s}(t) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t-s)},$$

which results in the following hereditary integral:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \operatorname{dev}\left[\int_0^t \dot{g}_R(s)\bar{\mathbf{F}}_t^{-1}(t-s) \cdot \boldsymbol{\tau}_0(t-s) \cdot \bar{\mathbf{F}}_t^{-T}(t-s)\, ds \right],$$

where $\boldsymbol{\tau}$ is the deviatoric part of the Kirchhoff stress.

The finite-strain theory is described in more detail in "Finite-strain viscoelasticity," Section 4.8.2 of the Abaqus Theory Manual.

Defining the volumetric behavior

The volumetric behavior can be written in a form that is similar to the shear behavior:

$$p(t) = -\mathrm{K}_0 \int_0^t k_R(t-s)\,\dot{\varepsilon}^{vol}(s)\,ds,$$

where p is the hydrostatic pressure, K_0 is the instantaneous elastic bulk modulus, $k_R(t)$ is the dimensionless bulk relaxation modulus, and ε^{vol} is the volume strain.

The above expansion applies to small as well as finite strain since the volume strains are generally small and there is no need to map the pressure from time $t-s$ to time t.

Defining viscoelastic behavior for traction-separation elasticity in Abaqus/Explicit

Time domain viscoelasticity can be used in Abaqus/Explicit to model rate-dependent behavior of cohesive elements with traction-separation elasticity ("Defining elasticity in terms of tractions and separations for cohesive elements" in "Linear elastic behavior," Section 22.2.1). In this case the evolution equation for the normal and two shear nominal tractions take the form:

$$t_n(t) = t_n^0(t) + \int_0^t \dot{k}_R(s) t_n^0(t-s)\,ds,$$

$$t_s(t) = t_s^0(t) + \int_0^t \dot{g}_R(s) t_s^0(t-s)\,ds,$$

$$t_t(t) = t_t^0(t) + \int_0^t \dot{g}_R(s) t_t^0(t-s)\,ds,$$

where $t_n^0(t)$, $t_s^0(t)$, and $t_t^0(t)$ are the instantaneous nominal tractions at time t in the normal and the two local shear directions, respectively. The functions $g_R(t)$ and $k_R(t)$ now represent the dimensionless shear and normal relaxation moduli, respectively. Note the close similarity between the viscoelastic formulation for the continuum elastic response discussed in the previous sections and the formulation for cohesive behavior with traction-separation elasticity after reinterpreting shear and bulk relaxation as shear and normal relaxation.

For the case of uncoupled traction elasticity, the viscoelastic normal and shear behaviors are assumed to be independent. The normal relaxation modulus is defined as

$$k_R(t) = K_{nn}(t)/K_{nn}^0,$$

where K_{nn}^0 is the instantaneous normal moduli. The shear relaxation modulus is assumed to be isotropic and, therefore, independent of the local shear directions:

$$g_R(t) = K_{ss}(t)/K_{ss}^0 = K_{tt}(t)/K_{tt}^0,$$

where K^0_{ss} and K^0_{tt} are the instantaneous shear moduli.

For the case of coupled traction-separation elasticity the normal and shear relaxation moduli must be the same, $g_R(t) = k_R(t)$, and you must use the same relaxation data for both behaviors.

Temperature effects

The effect of temperature, θ, on the material behavior is introduced through the dependence of the instantaneous stress, τ_0, on temperature and through a reduced time concept. The expression for the linear-elastic shear stress is rewritten as

$$\tau(t) = G_0(\theta) \int_0^t g_R\big(\xi(t) - \xi(s)\big)\dot{\gamma}(s)\, ds,$$

where the instantaneous shear modulus G_0 is temperature dependent and $\xi(t)$ is the reduced time, defined by

$$\xi(t) = \int_0^t \frac{ds}{A\big(\theta(s)\big)},$$

where $A\big(\theta(t)\big)$ is a shift function at time t. This reduced time concept for temperature dependence is usually referred to as thermo-rheologically simple (TRS) temperature dependence. Often the shift function is approximated by the Williams-Landel-Ferry (WLF) form. See "Thermo-rheologically simple temperature effects" below, for a description of the WLF and other forms of the shift function available in Abaqus.

The reduced time concept is also used for the volumetric behavior, the large-strain formulation, and the traction-separation formulation.

Numerical implementation

Abaqus assumes that the viscoelastic material is defined by a Prony series expansion of the dimensionless relaxation modulus:

$$g_R(t) = 1 - \sum_{i=1}^{N} \bar{g}_i^P (1 - e^{-t/\tau_i^G}),$$

where N, $\bar{g}_i^P$, and τ_i^G, $i = 1, 2, \ldots, N$, are material constants. For linear isotropic elasticity, substitution in the small-strain expression for the shear stress yields

$$\tau(t) = G_0 \left(\gamma - \sum_{i=1}^{N} \gamma_i \right),$$

where

$$\gamma_i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} \gamma(t-s)\, ds.$$

The γ_i are interpreted as state variables that control the stress relaxation, and

$$\gamma^{cr} = \sum_{i=1}^{N} \gamma_i$$

is the "creep" strain: the difference between the total mechanical strain and the instantaneous elastic strain (the stress divided by the instantaneous elastic modulus). In Abaqus/Standard γ^{cr} is available as the creep strain output variable CE ("Abaqus/Standard output variable identifiers," Section 4.2.1).

A similar Prony series expansion is used for the volumetric response, which is valid for both small- and finite-strain applications:

$$p = -K_0 \left(\varepsilon^{vol} - \sum_{i=1}^{N} \varepsilon_i^{vol} \right),$$

where

$$\varepsilon_i^{vol} = \frac{\bar{k}_i^P}{\tau_i^K} \int_0^t e^{-s/\tau_i^K} \varepsilon^{vol}(t-s)\, ds.$$

Abaqus assumes that $\tau_i^G = \tau_i^K = \tau_i$.

For linear anisotropic elasticity, the Prony series expansion, in combination with the generalized small-strain expression for the deviatoric stress, yields

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 - \sum_{i=1}^{N} \boldsymbol{\tau}_i,$$

where

$$\boldsymbol{\tau}_i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} \boldsymbol{\tau}_0(t-s)\, ds.$$

The $\boldsymbol{\tau}_i$ are interpreted as state variables that control the stress relaxation.

For finite strains, the Prony series expansion, in combination with the finite-strain expression for the shear stress, produces the following expression for the deviatoric stress:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 - \sum_{i=1}^{N} \mathrm{dev}\left(\boldsymbol{\tau}_i\right),$$

where

$$\boldsymbol{\tau}_i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} \bar{\mathbf{F}}_t^{-1}(t-s) \cdot \boldsymbol{\tau}_0(t-s) \cdot \bar{\mathbf{F}}_t(t-s)\, ds.$$

The $\boldsymbol{\tau}_i$ are interpreted as state variables that control the stress relaxation.

For traction-separation elasticity, the Prony series expansion yields

$$\mathbf{t} = \begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{Bmatrix} t_n^0 \\ t_s^0 \\ t_t^0 \end{Bmatrix} - \sum_{i=1}^{N} \begin{Bmatrix} t_n^i \\ t_s^i \\ t_t^i \end{Bmatrix} = \mathbf{t}^0 - \sum_{i=1}^{N} \mathbf{t}^i,$$

where

$$t_n^i = \frac{\bar{k}_i^P}{\tau_i^K} \int_0^t e^{-s/\tau_i^K} t_n^0(t-s)\, ds,$$

$$t_s^i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} t_s^0(t-s)\, ds,$$

$$t_t^i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} t_t^0(t-s)\, ds.$$

The $\mathbf{t}^i$ are interpreted as state variables that control the relaxation of the traction stresses.

If the instantaneous material behavior is defined by linear elasticity or hyperelasticity, the bulk and shear behavior can be defined independently. However, if the instantaneous behavior is defined by the hyperfoam model, the deviatoric and volumetric constitutive behavior are coupled and it is mandatory to use the same relaxation data for both behaviors. For linear anisotropic elasticity, the same relaxation data should be used for both behaviors when the elasticity definition is such that the deviatoric and volumetric response is coupled. Similarly, for coupled traction-separation elasticity you must use the same relaxation data for the normal and shear behaviors.

In all of the above expressions temperature dependence is readily introduced by replacing e^{-s/τ_i^G} by $e^{-\xi(s)/\tau_i^G}$ and e^{-s/τ_i^K} by $e^{-\xi(s)/\tau_i^K}$.

Determination of viscoelastic material parameters

The above equations are used to model the time-dependent shear and volumetric behavior of a viscoelastic material. The relaxation parameters can be defined in one of four ways: direct specification of the Prony series parameters, inclusion of creep test data, inclusion of relaxation test data, or inclusion of frequency-dependent data obtained from sinusoidal oscillation experiments. Temperature effects are included in the same manner regardless of the method used to define the viscoelastic material.

Abaqus/CAE allows you to evaluate the behavior of viscoelastic materials by automatically creating response curves based on experimental test data or coefficients. A viscoelastic material can be evaluated only if it is defined in the time domain and includes hyperelastic and/or elastic material data.

See "Evaluating hyperelastic and viscoelastic material behavior," Section 12.4.7 of the Abaqus/CAE User's Manual.

Direct specification

The Prony series parameters $\bar{g}_i^P$, $\bar{k}_i^P$, and τ_i can be defined directly for each term in the Prony series. There is no restriction on the number of terms that can be used. If a relaxation time is associated with only one of the two moduli, leave the other one blank or enter a zero. The data should be given in ascending order of the relaxation time. The number of lines of data given defines the number of terms in the Prony series, N. If this model is used in conjunction with the hyperfoam material model, the two modulus ratios have to be the same ($\bar{g}_i^P = \bar{k}_i^P$).

Input File Usage: *VISCOELASTIC, TIME=PRONY

The data line is repeated as often as needed to define the first, second, third, etc. terms in the Prony series.

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time** and **Time: Prony**

Enter as many rows of data in the table as needed to define the first, second, third, etc. terms in the Prony series.

Creep test data

If creep test data are specified, Abaqus will calculate the terms in the Prony series automatically. The normalized shear and bulk compliances are defined as

$$j_S(t) = G_0\, J_S(t) \qquad \text{and} \qquad j_K(t) = K_0\, J_K(t),$$

where $J_S(t) = \gamma(t)/\tau_0$ is the shear compliance, $\gamma(t)$ is the total shear strain, and τ_0 is the constant shear stress in a shear creep test; $J_K(t) = \varepsilon^{vol}(t)/p_0$ is the volumetric compliance, $\varepsilon^{vol}(t)$ is the total volumetric strain, and p_0 is the constant pressure in a volumetric creep test. At time $t = 0$, $j_S(0) = j_K(0) = 1$.

The creep data are converted to relaxation data through the convolution integrals

$$\int_0^t g_R(s) j_S(t-s)ds = t \qquad \text{and} \qquad \int_0^t k_R(s) j_K(t-s)ds = t.$$

Abaqus then uses the normalized shear modulus $g_R(t)$ and normalized bulk modulus $k_R(t)$ in a nonlinear least-squares fit to determine the Prony series parameters.

Using the shear and volumetric test data consecutively

The shear test data and volumetric test data can be used consecutively to define the normalized shear and bulk compliances as functions of time. A separate least-squares fit is performed on each data set; and the two derived sets of Prony series parameters, $(\bar{g}_i^P, \tau_i^G)$ and $(\bar{k}_i^P, \tau_i^K)$, are merged into one set of parameters, $(\bar{g}_i^P, \bar{k}_i^P, \tau_i)$.

Input File Usage: Use the following three options. The first option is required. Only one of the second and third options is required.

*VISCOELASTIC, TIME=CREEP TEST DATA
*SHEAR TEST DATA
*VOLUMETRIC TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time** and **Time: Creep test data**

In addition, select one or both of the following:

Test Data→Shear Test Data
Test Data→Volumetric Test Data

Using the combined test data

Alternatively, the combined test data can be used to specify the normalized shear and bulk compliances simultaneously as functions of time. A single least-squares fit is performed on the combined set of test data to determine one set of Prony series parameters, $(\bar{g}_i^P, \bar{k}_i^P, \tau_i)$.

Input File Usage: Use both of the following options:

*VISCOELASTIC, TIME=CREEP TEST DATA
*COMBINED TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time**, **Time: Creep test data**, and **Test Data→Combined Test Data**

Relaxation test data

As with creep test data, Abaqus will calculate the Prony series parameters automatically from relaxation test data.

Using the shear and volumetric test data consecutively

Again, the shear test data and volumetric test data can be used consecutively to define the relaxation moduli as functions of time. A separate nonlinear least-squares fit is performed on each data set; and the two derived sets of Prony series parameters, $(\bar{g}_i^P, \tau_i^G)$ and $(\bar{k}_i^P, \tau_i^K)$, are merged into one set of parameters, $(\bar{g}_i^P, \bar{k}_i^P, \tau_i)$.

Input File Usage: Use the following three options. The first option is required. Only one of the second and third options is required.

*VISCOELASTIC, TIME=RELAXATION TEST DATA
*SHEAR TEST DATA
*VOLUMETRIC TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time** and **Time: Relaxation test data**

In addition, select one or both of the following:

Test Data→Shear Test Data
Test Data→Volumetric Test Data

Using the combined test data

Alternatively, the combined test data can be used to specify the relaxation moduli simultaneously as functions of time. A single least-squares fit is performed on the combined set of test data to determine one set of Prony series parameters, $(\bar{g}_i^P, \bar{k}_i^P, \tau_i)$.

Input File Usage: Use both of the following options:

∗VISCOELASTIC, TIME=RELAXATION TEST DATA
∗COMBINED TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time**, **Time: Relaxation test data**, and **Test Data→Combined Test Data**

Frequency-dependent test data

The Prony series terms can also be calibrated using frequency-dependent test data. In this case Abaqus uses analytical expressions that relate the Prony series relaxation functions to the storage and loss moduli. The expressions for the shear moduli, obtained by converting the Prony series terms from the time domain to the frequency domain by making use of Fourier transforms, can be written as follows:

$$G_s(\omega) = G_0[1 - \sum_{i=1}^{N} \bar{g}_i^P] + G_0 \sum_{i=1}^{N} \frac{\bar{g}_i^P \tau_i^2 \omega^2}{1 + \tau_i^2 \omega^2},$$

$$G_\ell(\omega) = G_0 \sum_{i=1}^{N} \frac{\bar{g}_i^P \tau_i \omega}{1 + \tau_i^2 \omega^2},$$

where $G_s(\omega)$ is the storage modulus, $G_\ell(\omega)$ is the loss modulus, ω is the angular frequency, and N is the number of terms in the Prony series. These expressions are used in a nonlinear least-squares fit to determine the Prony series parameters from the storage and loss moduli cyclic test data obtained at M frequencies by minimizing the error function χ^2:

$$\chi^2 = \sum_{i=1}^{M} \frac{1}{G_\infty^2}[(G_s - \bar{G}_s)_i^2 + (G_\ell - \bar{G}_\ell)_i^2],$$

where $\bar{G}_s$ and $\bar{G}_\ell$ are the test data and G_0 and G_∞, respectively, are the instantaneous and long-term shear moduli. The expressions for the bulk moduli, $\mathrm{K}_s(\omega)$ and $\mathrm{K}_\ell(\omega)$, are written analogously.

The frequency domain data are defined in tabular form by giving the real and imaginary parts of ωg^* and ωk^*—where ω is the circular frequency—as functions of frequency in cycles per time. $g^*(\omega)$ is the Fourier transform of the nondimensional shear relaxation function $g(t) = \frac{G_R(t)}{G_\infty} - 1$. Given the frequency-dependent storage and loss moduli $G_s(\omega)$, $G_\ell(\omega)$, $\mathrm{K}_s(\omega)$, and $\mathrm{K}_\ell(\omega)$, the real and imaginary parts of ωg^* and ωk^* are then given as

$$\omega\Re(g^*) = G_\ell/G_\infty,\ \omega\Im(g*) = 1 - G_s/G_\infty, \quad \omega\Re(k^*) = \mathrm{K}_\ell/\mathrm{K}_\infty,\ \omega\Im(k^*) = 1 - \mathrm{K}_s/\mathrm{K}_\infty,$$

where G_∞ and K_∞ are the long-term shear and bulk moduli determined from the elastic or hyperelastic properties.

Input File Usage: *VISCOELASTIC, TIME=FREQUENCY DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time** and **Time: Frequency data**

Calibrating the Prony series parameters

You can specify two optional parameters related to the calibration of Prony series parameters for viscoelastic materials: the error tolerance and N_{max}. The error tolerance is the allowable average root-mean-square error of data points in the least-squares fit, and its default value is 0.01. N_{max} is the maximum number of terms N in the Prony series, and its default (and maximum) value is 13. Abaqus will perform the least-squares fit from $N = 1$ to $N = N_{max}$ until convergence is achieved for the lowest N with respect to the error tolerance.

The following are some guidelines for determining the number of terms in the Prony series from test data. Based on these guidelines, you can choose N_{max}.

- There should be enough data pairs for determining all the parameters in the Prony series terms. Thus, assuming that N is the number of Prony series terms, there should be a total of at least $2N$ data points in shear test data, $2N$ data points in volumetric test data, $3N$ data points in combined test data, and $2N$ data points in the frequency domain.
- The number of terms in the Prony series should be typically not more than the number of logarithmic "decades" spanned by the test data. The number of logarithmic "decades" is defined as $\log_{10}(t_{max}/t_{min})$, where t_{max} and t_{min} are the maximum and minimum time in the test data, respectively.

Input File Usage: *VISCOELASTIC, ERRTOL=*error_tolerance*, NMAX=N_{max}

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time**; **Time: Creep test data**, **Relaxation test data**, or **Frequency data**; **Maximum number of terms in the Prony series:** N_{max}; and **Allowable average root-mean-square error:** *error_tolerance*

Thermo-rheologically simple temperature effects

Regardless of the method used to define the viscoelastic behavior, thermo-rheologically simple temperature effects can be included by specifying the method used to define the shift function. Abaqus supports the following forms of the shift function: the Williams-Landel-Ferry (WLF) form, the Arrhenius form, and user-defined forms.

Thermo-rheologically simple temperature effects can also be included in the definition of equation of state models with viscous shear behavior (see "Viscous shear behavior" in "Equation of state," Section 25.2.1).

Williams-Landel-Ferry (WLF) form

The shift function can be defined by the Williams-Landel-Ferry (WLF) approximation, which takes the form:

$$\log_{10}(A) = -\frac{C_1(\theta - \theta_0)}{C_2 + (\theta - \theta_0)},$$

where θ_0 is the reference temperature at which the relaxation data are given; θ is the temperature of interest; and C_1, C_2 are calibration constants obtained at this temperature. If $\theta \leq \theta_0 - C_2$, deformation changes will be elastic, based on the instantaneous moduli.

For additional information on the WLF equation, see "Viscoelasticity," Section 4.8.1 of the Abaqus Theory Manual.

Input File Usage: *TRS, DEFINITION=WLF

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time**, **Time: *any method***, and **Suboptions→Trs**: **Shift function: WLF**

Arrhenius form

The Arrhenius shift function is commonly used for semi-crystalline polymers. It takes the form

$$\ln(A) = \frac{E_0}{R}\left(\frac{1}{\theta - \theta^Z} - \frac{1}{\theta_0 - \theta^Z}\right),$$

where E_0 is the activation energy, R is the universal gas constant, θ^Z is the absolute zero in the temperature scale being used, θ_0 is the reference temperature at which the relaxation data are given, and θ is the temperature of interest.

Input File Usage: Use the following option to define the Arrhenius shift function:

*TRS, DEFINITION=ARRHENIUS

In addition, use the *PHYSICAL CONSTANTS option to specify the universal gas constant and absolute zero.

Abaqus/CAE Usage: The Arrhenius shift function is not supported in Abaqus/CAE.

User-defined form

The shift function can be specified alternatively in user subroutines **`UTRS`** in Abaqus/Standard and **`VUTRS`** in Abaqus/Explicit.

Input File Usage: *TRS, DEFINITION=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Time**, **Time: *any method***, and **Suboptions→Trs**: **Shift function: User subroutine UTRS**

Defining the rate-independent part of the material response

In all cases elastic moduli must be specified to define the rate-independent part of the material behavior. Small-strain linear elastic behavior is defined by an elastic material model ("Linear elastic behavior," Section 22.2.1), and large-deformation behavior is defined by a hyperelastic ("Hyperelastic behavior of rubberlike materials," Section 22.5.1) or hyperfoam ("Hyperelastic behavior in elastomeric foams," Section 22.5.2) material model. The rate-independent elasticity for any of these models can be defined in terms of either instantaneous elastic moduli or long-term elastic moduli. The choice of defining the elasticity in terms of instantaneous or long-term moduli is a matter of convenience only; it does not have an effect on the solution.

The effective relaxation moduli are obtained by multiplying the instantaneous elastic moduli with the dimensionless relaxation functions as described below.

Linear elastic isotropic materials

For linear elastic isotropic material behavior

$$G_R(t) = G_0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k}) \right)$$

and

$$\mathrm{K}_R(t) = \mathrm{K}_0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k}) \right),$$

where G_0 and K_0 are the instantaneous shear and bulk moduli determined from the values of the user-defined instantaneous elastic moduli E_0 and ν_0: $G_0 = E_0/2(1+\nu_0)$ and $\mathrm{K}_0 = E_0/3(1-2\nu_0)$.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$G_\infty = G_0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P \right) \qquad \text{and} \qquad K_\infty = K_0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P \right).$$

Linear elastic anisotropic materials

For linear elastic anisotropic material behavior the relaxation coefficients are applied to the elastic moduli as

$$\overline{\mathbf{D}}_R(t) = \overline{\mathbf{D}}_0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k}) \right)$$

and

$$\mathrm{K}_R(t) = \mathrm{K}_0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k}) \right),$$

where $\overline{\mathbf{D}}_0$ and K_0 are the instantaneous deviatoric elasticity tensor and bulk moduli determined from the values of the user-defined instantaneous elastic moduli $\mathbf{D}_0$. If both shear and bulk relaxation coefficients are specified and they are unequal, Abaqus issues an error message if the elastic moduli $\mathbf{D}_0$ is such that the deviatoric and volumetric response is coupled.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$\overline{\mathbf{D}}_\infty = \overline{\mathbf{D}}_0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P \right) \qquad \text{and} \qquad K_\infty = K_0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P \right).$$

Hyperelastic materials

For hyperelastic material behavior the relaxation coefficients are applied either to the constants that define the energy function or directly to the energy function. For the polynomial function and its particular cases (reduced polynomial, Mooney-Rivlin, neo-Hookean, and Yeoh)

$$C_{ij}^R(t) = C_{ij}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k}) \right),$$

for the Ogden function

$$\mu_i^R(t) = \mu_i^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k}) \right),$$

for the Arruda-Boyce and Van der Waals functions

$$\mu^R(t) = \mu^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k}) \right),$$

and for the Marlow function

$$U_{dev}^R(t) = U_{dev}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k}) \right).$$

For the coefficients governing the compressible behavior of the polynomial models and the Ogden model

$$D_i^R(t) = D_i^0 / \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k}) \right),$$

for the Arruda-Boyce and Van der Waals functions

$$D^R(t) = D^0 / \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k})\right),$$

and for the Marlow function

$$U_{vol}^R(t) = U_{vol}^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k})\right).$$

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$C_{ij}^\infty = C_{ij}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right), \quad \text{or} \quad \mu_i^\infty = \mu_i^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right), \quad \text{or} \quad \mu^\infty = \mu^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right),$$

while the instantaneous bulk compliance moduli are obtained from

$$D_i^\infty = D_i^0 / \left(1 - \sum_{k=1}^{N} \bar{k}_k^P\right), \quad \text{or} \quad D^\infty = D^0 / \left(1 - \sum_{k=1}^{N} \bar{k}_k^P\right);$$

for the Marlow functions we have

$$U_{dev}^\infty = U_{dev}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right) \quad \text{and} \quad U_{vol}^\infty = U_{vol}^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P\right).$$

Mullins effect

If long-term moduli are defined for the underlying hyperelastic behavior, the instantaneous value of the parameter m in Mullins effect is determined from

$$m^\infty = m^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right).$$

Elastomeric foams

For elastomeric foam material behavior the instantaneous shear and bulk relaxation coefficients are assumed to be equal and are applied to the material constants μ_i in the energy function:

$$\mu_i^R(t) = \mu_i^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k})\right) = \mu_i^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k})\right).$$

If only the shear relaxation coefficients are specified, the bulk relaxation coefficients are set equal to the shear relaxation coefficients and vice versa. If both shear and bulk relaxation coefficients are specified and they are unequal, Abaqus issues an error message.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$\mu_i^\infty = \mu_i^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right) = \mu_i^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P\right).$$

Traction-separation elasticity

For cohesive elements with uncoupled traction-separation elastic behavior:

$$K_{nn}(t) = K_{nn}^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k})\right),$$

$$K_{ss}(t) = K_{ss}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k})\right),$$

and

$$K_{tt}(t) = K_{tt}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k})\right),$$

where K_{nn}^0 is the instantaneous normal modulus and K_{ss}^0 and K_{tt}^0 are the instantaneous shear moduli. If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$K_{nn}^\infty / K_{nn}^0 = \left(1 - \sum_{k=1}^{N} \bar{k}_k^P\right), \qquad \text{and} \qquad K_{ss}^\infty / K_{ss}^0 = K_{tt}^\infty / K_{tt}^0 = \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right).$$

For cohesive elements with coupled traction-separation elastic behavior the shear and bulk relaxation coefficients must be equal:

$$\mathbf{K}(t) = \mathbf{K}^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P (1 - e^{-t/\tau_k})\right) = \mathbf{K}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P (1 - e^{-t/\tau_k})\right),$$

where $\mathbf{K}^0$ is the user-defined instantaneous elasticity matrix. If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$\mathbf{K}^\infty = \mathbf{K}^0 \left(1 - \sum_{k=1}^{N} \bar{k}_k^P\right) = \mathbf{K}^0 \left(1 - \sum_{k=1}^{N} \bar{g}_k^P\right).$$

Material response in different analysis procedures

The time-domain viscoelastic material model is active during the following procedures:

- transient static analysis ("Quasi-static analysis," Section 6.2.5),
- transient implicit dynamic analysis ("Implicit dynamic analysis using direct integration," Section 6.3.2),
- explicit dynamic analysis ("Explicit dynamic analysis," Section 6.3.3),
- steady-state transport analysis ("Steady-state transport analysis," Section 6.4.1),
- fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3),
- fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4), and
- transient (consolidation) coupled pore fluid diffusion and stress analysis ("Coupled pore fluid diffusion and stress analysis," Section 6.8.1).

Viscoelastic material response is always ignored in a static analysis. It can also be ignored in a coupled temperature-displacement analysis, a coupled thermal-electrical-structural analysis, or a soils consolidation analysis by specifying that no creep or viscoelastic response is occurring during the step even if creep or viscoelastic material properties are defined (see "Fully coupled thermal-stress analysis," Section 6.5.3, or "Coupled pore fluid diffusion and stress analysis," Section 6.8.1). In these cases it is assumed that the loading is applied instantaneously, so that the resulting response corresponds to an elastic solution based on instantaneous elastic moduli.

Abaqus/Standard also provides the option to obtain the fully relaxed long-term elastic solution directly in a static or steady-state transport analysis without having to perform a transient analysis. The long-term value is used for this purpose. The viscous damping stresses (the internal stresses associated with each of the Prony-series terms) are increased gradually from their values at the beginning of the step to their long-term values at the end of the step if the long-term value is specified.

Use with other material models

The viscoelastic material model must be combined with an elastic material model. It is used with the isotropic linear elasticity model ("Linear elastic behavior," Section 22.2.1) to define classical, linear, small-strain, viscoelastic behavior or with the hyperelastic ("Hyperelastic behavior of rubberlike materials," Section 22.5.1) or hyperfoam ("Hyperelastic behavior in elastomeric foams," Section 22.5.2) models to define large-deformation, nonlinear, viscoelastic behavior. It can also be used with anisotropic linear elasticity and with traction-separation elastic behavior in Abaqus/Explicit. The elastic properties defined for these models can be temperature dependent.

Viscoelasticity cannot be combined with any of the plasticity models. See "Combining material behaviors," Section 21.1.3, for more details.

Elements

The time domain viscoelastic material model can be used with any stress/displacement, coupled temperature-displacement, or thermal-electrical-structural element in Abaqus.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning in Abaqus/Standard if viscoelasticity is defined:

EE	Elastic strain corresponding to the stress state at time t and the instantaneous elastic material properties.
CE	Equivalent creep strain defined as the difference between the total strain and the elastic strain.

Considerations for steady-state transport analysis

When a steady-state transport analysis ("Steady-state transport analysis," Section 6.4.1) is combined with large-strain viscoelasticity, the viscous dissipation, CENER, is computed as the energy dissipated per revolution as a material point is transported around its streamline; that is,

$$W_{cr} = \oint \boldsymbol{\sigma} : d\boldsymbol{\varepsilon}.$$

Consequently, all the material points in a given streamline report the same value for CENER, and other derived quantities such as ELCD and ALLCD also have the meaning of dissipation per revolution. The recoverable elastic strain energy density, SENER, is approximated as

$$W_{el} = W_{el}^{t} + W_{cr}^{t} + \Delta W - W_{cr},$$

where ΔW is the incremental energy input and t is the time at the beginning of the current increment. Since two different units are used in the quantities appearing in the above equation, a proper meaning cannot be assigned to quantities such as SENER, ELSE, ALLSE, and ALLIE.

Considerations for large-strain viscoelasticity in Abaqus/Explicit

For the case of large-strain viscoelasticity, Abaqus/Explicit does not compute the viscous dissipation for performance reasons. Instead, the contribution of viscous dissipation is included in the strain energy output, SENER; and CENER is output as zero. Consequently, special care must be exercised when interpreting strain energy results of large-strain viscoelastic materials in Abaqus/Explicit since they include viscous dissipation effects.

22.7.2 FREQUENCY DOMAIN VISCOELASTICITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "Time domain viscoelasticity," Section 22.7.1
- *VISCOELASTIC
- "Defining frequency domain viscoelasticity" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The frequency domain viscoelastic material model:

- describes frequency-dependent material behavior in small steady-state harmonic oscillations for those materials in which dissipative losses caused by "viscous" (internal damping) effects must be modeled in the frequency domain;
- assumes that the shear (deviatoric) and volumetric behaviors are independent in multiaxial stress states;
- can be used in large-strain problems;
- can be used only in conjunction with "Linear elastic behavior," Section 22.2.1; "Hyperelastic behavior of rubberlike materials," Section 22.5.1; and "Hyperelastic behavior in elastomeric foams," Section 22.5.2, to define the long-term elastic material properties;
- can be used in conjunction with the elastic-damage gasket behavior ("Defining a nonlinear elastic model with damage" in "Defining the gasket behavior directly using a gasket behavior model," Section 32.6.6) to define the effective thickness-direction storage and loss moduli for gasket elements; and
- is active only during the direct-solution steady-state dynamic ("Direct-solution steady-state dynamic analysis," Section 6.3.4), the subspace-based steady-state dynamic ("Subspace-based steady-state dynamic analysis," Section 6.3.9), the natural frequency extraction ("Natural frequency extraction," Section 6.3.5), and the complex eigenvalue extraction ("Complex eigenvalue extraction," Section 6.3.6) procedures.

Defining the shear behavior

Consider a shear test at small strain, in which a harmonically varying shear strain γ is applied:

$$\gamma(t) = \gamma_0 \exp(i\omega t),$$

where γ_0 is the amplitude, $i = \sqrt{-1}$, ω is the circular frequency, and t is time. We assume that the specimen has been oscillating for a very long time so that a steady-state solution is obtained. The solution for the shear stress then has the form

$$\tau(t) = \big(G_s(\omega) + iG_\ell(\omega)\big)\gamma_0 \exp(i\omega t),$$

where G_s and G_ℓ are the shear storage and loss moduli. These moduli can be expressed in terms of the (complex) Fourier transform $g^*(\omega)$ of the nondimensional shear relaxation function $g(t) = \frac{G_R(t)}{G_\infty} - 1$:

$$G_s(\omega) = G_\infty\big(1 - \omega\Im(g^*)\big), \qquad G_\ell(\omega) = G_\infty\big(\omega\Re(g^*)\big),$$

where $G_R(t)$ is the time-dependent shear relaxation modulus, $\Re(g^*)$ and $\Im(g^*)$ are the real and imaginary parts of $g^*(\omega)$, and G_∞ is the long-term shear modulus. See "Frequency domain viscoelasticity," Section 4.8.3 of the Abaqus Theory Manual, for details.

The above equation states that the material responds to steady-state harmonic strain with a stress of magnitude $G_s\gamma_0$ that is in phase with the strain and a stress of magnitude $G_\ell\gamma_0$ that lags the excitation by $90°$. Hence, we can regard the factor

$$G^*(\omega) = G_s(\omega) + iG_\ell(\omega)$$

as the complex, frequency-dependent shear modulus of the steadily vibrating material. The absolute magnitude of the stress response is

$$|\tau| = \sqrt{G_s^2(\omega) + G_\ell^2(\omega)}\ |\gamma_0|,$$

and the phase lag of the stress response is

$$\phi = \arctan\left(\frac{G_\ell(\omega)}{G_s(\omega)}\right).$$

Measurements of $\tau|$ and ϕ as functions of frequency in an experiment can, thus, be used to define G_s and G_ℓ and, thus, $\Re(g^*)$ and $\Im(g^*)$ as functions of frequency.

Unless stated otherwise explicitly, all modulus measurements are assumed to be "true" quantities.

Defining the volumetric behavior

In multiaxial stress states Abaqus/Standard assumes that the frequency dependence of the shear (deviatoric) and volumetric behaviors are independent. The volumetric behavior is defined by the bulk storage and loss moduli $\mathrm{K}_s(\omega)$ and $\mathrm{K}_\ell(\omega)$. Similar to the shear moduli, these moduli can also be expressed in terms of the (complex) Fourier transform $k^*(\omega)$ of the nondimensional bulk relaxation function $k(t)$:

$$K_s(w) = K_\infty\big(1 - \omega\Im(k^*)\big), \qquad K_\ell(w) = K_\infty\big(\omega\Re(k^*)\big),$$

where K_∞ is the long-term elastic bulk modulus.

Large-strain viscoelasticity

The linearized vibrations can also be associated with an elastomeric material whose long-term (elastic) response is nonlinear and involves finite strains (a hyperelastic material). We can retain the simplicity of the steady-state small-amplitude vibration response analysis in this case by assuming that the linear expression for the shear stress still governs the system, except that now the long-term shear modulus G_∞ can vary with the amount of static prestrain $\bar{\gamma}$:

$$G_\infty = G_\infty(\bar{\gamma}).$$

The essential simplification implied by this assumption is that the frequency-dependent part of the material's response, defined by the Fourier transform $g^*(\omega)$ of the relaxation function, is not affected by the magnitude of the prestrain. Thus, strain and frequency effects are separated, which is a reasonable approximation for many materials.

Another implication of the above assumption is that the anisotropy of the viscoelastic moduli has the same strain dependence as the anisotropy of the long-term elastic moduli. Hence, the viscoelastic behavior in all deformed states can be characterized by measuring the (isotropic) viscoelastic moduli in the undeformed state.

In situations where the above assumptions are not reasonable, the data can be specified based on measurements at the prestrain level about which the steady-state dynamic response is desired. In this case you must measure G_s, G_ℓ, and G_∞ (likewise K_s, K_ℓ, and K_∞) at the prestrain level of interest. Alternatively, the viscoelastic data can be given directly in terms of uniaxial and volumetric storage and loss moduli that may be specified as functions of frequency and prestrain (see "Direct specification of storage and loss moduli for large-strain viscoelasticity" below.)

The generalization of these concepts to arbitrary three-dimensional deformations is provided in Abaqus/Standard by assuming that the frequency-dependent material behavior has two independent components: one associated with shear (deviatoric) straining and the other associated with volumetric straining. In the general case of a compressible material, the model is, therefore, defined for kinematically small perturbations about a predeformed state as

$$\frac{1}{J}\Delta^\nabla(J\mathbf{S}) = (1 + i\omega g^*)\mathbf{C}^S|_0 : \Delta\mathbf{e} + \mathbf{Q}|_0\Delta\varepsilon^{\text{vol}},$$

and

$$\Delta p = -\mathbf{Q}|_0 : \Delta\mathbf{e} - (1 + i\omega k^*)K|_0\Delta\varepsilon^{\text{vol}},$$

where

$\mathbf{S}$	is the deviatoric stress, $\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}$;
$\boldsymbol{p}$	is the equivalent pressure stress, $p = -\frac{1}{3}\text{trace}(\boldsymbol{\sigma})$;
$\Delta^\nabla(J\mathbf{S})$	is the part of the stress increment caused by incremental straining (as distinct from the part of the stress increment caused by incremental rotation of the preexisting stress with respect to the coordinate system);

J	is the ratio of current to original volume;
$\Delta\mathbf{e}$	is the (small) incremental deviatoric strain, $\Delta\mathbf{e} = \Delta\boldsymbol{\varepsilon} - \frac{1}{3}\Delta\varepsilon^{\text{vol}}\,\mathbf{I}$;
$\dot{\mathbf{e}}$	is the deviatoric strain rate, $\dot{\mathbf{e}} = \dot{\boldsymbol{\varepsilon}} - \frac{1}{3}\dot{\varepsilon}^{\text{vol}}\,\mathbf{I}$;
$\Delta\varepsilon^{\text{vol}}$	is the (small) incremental volumetric strain, $\Delta\varepsilon^{\text{vol}} = \text{trace}(\Delta\boldsymbol{\varepsilon})$;
$\dot{\varepsilon}^{\text{vol}}$	is the rate of volumetric strain, $\dot{\varepsilon}^{\text{vol}} = \text{trace}(\dot{\boldsymbol{\varepsilon}})$;
$\mathbf{C}^S\vert_0$	is the deviatoric tangent elasticity matrix of the material in its predeformed state (for example, C_{1212} is the tangent shear modulus of the prestrained material);
$\mathbf{Q}\vert_0$	is the volumetric strain-rate/deviatoric stress-rate tangent elasticity matrix of the material in its predeformed state; and
$K\vert_0$	is the tangent bulk modulus of the predeformed material.

For a fully incompressible material only the deviatoric terms in the first constitutive equation above remain and the viscoelastic behavior is completely defined by $g^*(\omega)$.

Determination of viscoelastic material parameters

The dissipative part of the material behavior is defined by giving the real and imaginary parts of g^* and k^* (for compressible materials) as functions of frequency. The moduli can be defined as functions of the frequency in one of three ways: by a power law, by tabular input, or by a Prony series expression for the shear and bulk relaxation moduli.

Power law frequency dependence

The frequency dependence can be defined by the power law formulæ

$$g^*(\omega) = g_1^* f^{-a} \qquad \text{and} \qquad k^*(\omega) = k_1^* f^{-b},$$

where a and b are real constants, g_1^* and k_1^* are complex constants, and $f = \omega/2\pi$ is the frequency in cycles per time.

Input File Usage: *VISCOELASTIC, FREQUENCY=FORMULA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Frequency** and **Frequency: Formula**

Tabular frequency dependence

The frequency domain response can alternatively be defined in tabular form by giving the real and imaginary parts of ωg^* and ωk^*—where ω is the circular frequency—as functions of frequency in cycles per time. Given the frequency-dependent storage and loss moduli $G_s(\omega)$, $G_\ell(\omega)$, $\mathrm{K}_s(\omega)$, and $\mathrm{K}_\ell(\omega)$, the real and imaginary parts of ωg^* and ωk^* are then given as

$$\omega\Re(g^*) = G_\ell/G_\infty, \;\; \omega\Im(g*) = 1 - G_s/G_\infty, \quad \omega\Re(k^*) = \mathrm{K}_\ell/\mathrm{K}_\infty, \;\; \omega\Im(k^*) = 1 - \mathrm{K}_s/\mathrm{K}_\infty,$$

where G_∞ and K_∞ are the long-term shear and bulk moduli determined from the elastic or hyperelastic properties.

Input File Usage: *VISCOELASTIC, FREQUENCY=TABULAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Frequency** and **Frequency: Tabular**

Abaqus provides an alternative approach for specifying the viscoelastic properties of hyperelastic and hyperfoam materials. This approach involves the direct (tabular) specification of storage and loss moduli from uniaxial and volumetric tests, as functions of excitation frequency and a measure of the level of pre-strain. The level of pre-strain refers to the level of elastic deformation at the base state about which the steady-state harmonic response is desired. This approach is discussed in "Direct specification of storage and loss moduli for large-strain viscoelasticity" below.

Prony series parameters

The frequency dependence can also be obtained from a time domain Prony series description of the dimensionless shear and bulk relaxation moduli:

$$g_R(t) = 1 - \sum_{i=1}^{N} \bar{g}_i^P (1 - e^{-t/\tau_i}),$$

$$k_R(t) = 1 - \sum_{i=1}^{N} \bar{k}_i^P (1 - e^{-t/\tau_i}),$$

where N, $\bar{g}_i^P$, $\bar{k}_i^P$, and τ_i, $i = 1, 2, \ldots, N$, are material constants. Using Fourier transforms, the expression for the time-dependent shear modulus can be written in the frequency domain as follows:

$$G_s(\omega) = G_0[1 - \sum_{i=1}^{N} \bar{g}_i^P] + G_0 \sum_{i=1}^{N} \frac{\bar{g}_i^P \tau_i^2 \omega^2}{1 + \tau_i^2 \omega^2},$$

$$G_\ell(\omega) = G_0 \sum_{i=1}^{N} \frac{\bar{g}_i^P \tau_i \omega}{1 + \tau_i^2 \omega^2},$$

where $G_s(\omega)$ is the storage modulus, $G_\ell(\omega)$ is the loss modulus, ω is the angular frequency, and N is the number of terms in the Prony series. The expressions for the bulk moduli, $\mathrm{K}_s(\omega)$ and $\mathrm{K}_\ell(\omega)$, are written analogously. Abaqus/Standard will automatically perform the conversion from the time domain to the frequency domain. The Prony series parameters $\bar{g}_i^P, \bar{k}_i^P, \tau_i$ can be defined in one of three ways: direct specification of the Prony series parameters, inclusion of creep test data, or inclusion of relaxation test data. If creep test data or relaxation test data are specified, Abaqus/Standard will determine the Prony series parameters in a nonlinear least-squares fit. A detailed description of the calibration of Prony series terms is provided in "Time domain viscoelasticity," Section 22.7.1.

For the test data you can specify the normalized shear and bulk data separately as functions of time or specify the normalized shear and bulk data simultaneously. A nonlinear least-squares fit is performed to determine the Prony series parameters, $(\bar{g}_i^P, \bar{k}_i^P, \tau_i)$.

Input File Usage: Use one of the following options to specify Prony data, creep test data, or relaxation test data:

*VISCOELASTIC, FREQUENCY=PRONY
*VISCOELASTIC, FREQUENCY=CREEP TEST DATA
*VISCOELASTIC, FREQUENCY=RELAXATION TEST DATA

Use one or both of the following options to specify the normalized shear and bulk data separately as functions of time:

*SHEAR TEST DATA
*VOLUMETRIC TEST DATA

Use the following option to specify the normalized shear and bulk data simultaneously:

*COMBINED TEST DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Frequency** and **Frequency: Prony**, **Creep test data**, or **Relaxation test data**

Use one or both of the following options to specify the normalized shear and bulk data separately as functions of time:

Test Data→Shear Test Data
Test Data→Volumetric Test Data

Use the following option to specify the normalized shear and bulk data simultaneously:

Test Data→Combined Test Data

Conversion of frequency-dependent elastic moduli

For some cases of small straining of isotropic viscoelastic materials, the material data are provided as frequency-dependent uniaxial storage and loss moduli, $E_s(\omega)$ and $E_\ell(\omega)$, and bulk moduli, $\mathrm{K}_s(\omega)$ and $\mathrm{K}_\ell(\omega)$. In that case the data must be converted to obtain the frequency-dependent shear storage and loss moduli $G_s(\omega)$ and $G_\ell(\omega)$.

The complex shear modulus is obtained as a function of the complex uniaxial and bulk moduli with the expression

$$G^* = \frac{3\mathrm{K}^* E^*}{9\mathrm{K}^* - E^*}.$$

Replacing the complex moduli by the appropriate storage and loss moduli, this expression transforms into

$$G_s + iG_\ell = \frac{3(\mathrm{K}_s + i\mathrm{K}_\ell)(E_s + iE_\ell)}{9(\mathrm{K}_s + i\mathrm{K}_\ell) - E_s + iE_\ell}.$$

After some algebra one obtains

$$G_s = 3\frac{9E_s(\mathrm{K}_s^2+\mathrm{K}_\ell^2)-\mathrm{K}_s(E_s^2+E_\ell^2)}{(9\mathrm{K}_s-E_s)^2+(9\mathrm{K}_\ell-E_\ell)^2}, \quad G_\ell = 3\frac{9E_\ell(\mathrm{K}_s^2+\mathrm{K}_\ell^2)-\mathrm{K}_\ell(E_s^2+E_\ell^2)}{(9\mathrm{K}_s-E_s)^2+(9\mathrm{K}_\ell-E_\ell)^2}.$$

Shear strain only

In many cases the viscous behavior is associated only with deviatoric straining, so that the bulk modulus is real and constant: $\mathrm{K}_s = \mathrm{K}_\infty$ and $\mathrm{K}_\ell = 0$. For this case the expressions for the shear moduli simplify to

$$G_s = 3\mathrm{K}_\infty\frac{9E_s\mathrm{K}_\infty - E_s^2 - E_\ell^2}{(9\mathrm{K}_\infty - E_s)^2 + E_\ell^2}, \qquad G_\ell = 3\mathrm{K}_\infty\frac{9E_\ell\mathrm{K}_\infty}{(9\mathrm{K}_\infty - E_s)^2 + E_\ell^2}.$$

Incompressible materials

If the bulk modulus is very large compared to the shear modulus, the material can be considered to be incompressible and the expressions simplify further to

$$G_s = E_s/3, \qquad G_\ell = E_\ell/3.$$

Direct specification of storage and loss moduli for large-strain viscoelasticity

For large-strain viscoelasticity Abaqus allows direct specification of storage and loss moduli from uniaxial and volumetric tests. This approach can be used when the assumption of the independence of viscoelastic properties on the pre-strain level is too restrictive.

You specify the storage and loss moduli directly as tabular functions of frequency, and you specify the level of pre-strain at the base state about which the steady-state dynamic response is desired. For uniaxial test data the measure of pre-strain is the uniaxial nominal strain; for volumetric test data the measure of pre-strain is the volume ratio. Abaqus internally converts the data that you specify to ratios of shear/bulk storage and loss moduli to the corresponding long-term elastic moduli. Subsequently, the basic formulation described in "Large-strain viscoelasticity" above is used.

For a general three-dimensional stress state it is assumed that the deviatoric part of the viscoelastic response depends on the level of pre-strain through the first invariant of the deviatoric left Cauchy-Green strain tensor (see "Hyperelastic material behavior," Section 4.6.1 of the Abaqus Theory Manual, for a definition of this quantity), while the volumetric part depends on the pre-strain through the volume ratio. A consequence of these assumptions is that for the uniaxial case, data can be specified from a uniaxial-tension preload state or from a uniaxial-compression preload state but not both.

The storage and loss moduli that you specify are assumed to be nominal quantities.

Input File Usage: Use the following option to specify only the uniaxial storage and loss moduli:

*VISCOELASTIC, PRELOAD=UNIAXIAL

You can also use the following option to specify the volumetric (bulk) storage and loss moduli:

*VISCOELASTIC, PRELOAD=VOLUMETRIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Viscoelastic**: **Domain: Frequency** and **Frequency: Tabular**

Use the following option to specify only the uniaxial storage and loss moduli:

Type: Isotropic or **Traction**: **Preload: Uniaxial**

Use the following option to specify only the volumetric storage and loss moduli:

Type: Isotropic: **Preload: Volumetric**

Use the following option to specify both uniaxial and volumetric moduli:

Type: Isotropic: **Preload: Uniaxial and Volumetric**

Defining the rate-independent part of the material behavior

In all cases elastic moduli must be specified to define the rate-independent part of the material behavior. The elastic behavior is defined by an elastic, hyperelastic, or hyperfoam material model. Since the frequency domain viscoelastic material model is developed around the long-term elastic moduli, the rate-independent elasticity must be defined in terms of long-term elastic moduli. This implies that the response in any analysis procedure other than a direct-solution steady-state dynamic analysis (such as a static preloading analysis) corresponds to the fully relaxed long-term elastic solution.

Use with other material models

The viscoelastic material model must be combined with the isotropic linear elasticity model to define classical, linear, small-strain, viscoelastic behavior. It is combined with the hyperelastic or hyperfoam model to define large-deformation, nonlinear, viscoelastic behavior. The long-term elastic properties defined for these models can be temperature dependent.

Viscoelasticity cannot be combined with any of the plasticity models. See “Combining material behaviors,” Section 21.1.3, for more details.

Elements

The frequency domain viscoelastic material model can be used with any stress/displacement element in Abaqus/Standard.

22.8 Nonlinear viscoelasticity

- "Hysteresis in elastomers," Section 22.8.1
- "Parallel network viscoelastic model," Section 22.8.2

22.8.1 HYSTERESIS IN ELASTOMERS

Products: Abaqus/Standard Abaqus/CAE

References

- "Elastic behavior: overview," Section 22.1.1
- *HYSTERESIS
- "Defining hysteretic behavior for an isotropic hyperelastic material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The hysteresis material model:

- defines strain-rate-dependent, hysteretic behavior of materials that undergo comparable elastic and inelastic strains;
- provides inelastic response only for shear distortional behavior—the response to volumetric deformations is purely elastic;
- can be used only in conjunction with "Hyperelastic behavior of rubberlike materials," Section 22.5.1, to define the elastic response of the material—the elasticity can be defined either in terms of the instantaneous moduli or the long-term moduli;
- is active during a static analysis ("Static stress analysis," Section 6.2.2), a quasi-static analysis ("Quasi-static analysis," Section 6.2.5), or a transient dynamic analysis using direct integration ("Implicit dynamic analysis using direct integration," Section 6.3.2)—it cannot be used in fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3), fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4), or steady-state transport analysis ("Steady-state transport analysis," Section 6.4.1);
- cannot be used to model temperature-dependent creep material properties—however, the elastic material properties can be temperature dependent; and
- uses unsymmetric matrix storage and solution by default.

Strain-rate-dependent material behavior for elastomers

Nonlinear strain-rate dependence of elastomers is modeled by decomposing the mechanical response into that of an equilibrium network (A) corresponding to the state that is approached in long-time stress relaxation tests and that of a time-dependent network (B) that captures the nonlinear rate-dependent deviation from the equilibrium state. The total stress is assumed to be the sum of the stresses in the two networks. The deformation gradient, $\mathbf{F}$, is assumed to act on both networks and is decomposed into elastic and inelastic parts in network B according to the multiplicative decomposition $\mathbf{F} = \mathbf{F}_B^e \cdot \mathbf{F}_B^{cr}$. The nonlinear rate-dependent material model is capable of reproducing the hysteretic behavior of elastomers

subjected to repeated cyclic loading. It does not model "Mullins effect"—the initial softening of an elastomer when it is first subjected to a load.

The material model is defined completely by:

- a hyperelastic material model that characterizes the elastic response of the model;
- a stress scaling factor, S, that defines the ratio of the stress carried by network B to the stress carried by network A under instantaneous loading; i.e., identical elastic stretching in both networks;
- a positive exponent, m, generally greater than 1, characterizing the effective stress dependence of the effective creep strain rate in network B;
- an exponent, C, restricted to lie in $[-1, 0]$, characterizing the creep strain dependence of the effective creep strain rate in network B;
- a nonnegative constant, A, in the expression for the effective creep strain rate—this constant also maintains dimensional consistency in the equation; and
- a constant, E, in the expression for the effective creep strain rate—this constant regularizes the creep strain rate near the undeformed state.

The effective creep strain rate in network B is given by the expression

$$\dot{\epsilon}_B^{cr} = A[\lambda_B^{cr} - 1 + E]^C(\sigma_B)^m,$$

where $\dot{\epsilon}_B^{cr}$ is the effective creep strain rate in network B, $\lambda_B^{cr} - 1$ is the nominal creep strain in network B, and σ_B is the effective stress in network B. The chain stretch in network B, λ_B^{cr}, is defined as

$$\lambda_B^{cr} = \sqrt{\frac{1}{3}\mathbf{I} : \mathbf{C}_B^{cr}},$$

where $\mathbf{C}_B^{cr} = \mathbf{F}_B^{cr\,T} \cdot \mathbf{F}_B^{cr}$. The effective stress in network B is defined as $\sigma_B = \sqrt{\frac{3}{2}\mathbf{S}_B : \mathbf{S}_B}$, where $\mathbf{S}_B$ is the deviatoric Cauchy stress tensor.

Defining strain-rate-dependent material behavior for elastomers

The elasticity of the model is defined by a hyperelastic material model. You input the stress scaling factor and the creep parameters for network B directly when you define the hysteresis material model. Typical values of the material parameters for a common elastomer are $S = 1.6$, $A = \frac{5}{(\sqrt{3})^m}(\text{sec})^{-1}(\text{MPa})^{-m}$, $m = 4$, $C = -1.0$, and $E = 0.01$ (Bergstrom and Boyce, 1998; 2001).

Input File Usage: Use both of the following options within the same material data block:

*HYSTERESIS
*HYPERELASTIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Hyperelastic**: **Suboptions→Hysteresis**

The input of the parameter E is not supported in Abaqus/CAE.

Elements

The use of the hysteresis material model is restricted to elements that can be used with hyperelastic materials (“Hyperelastic behavior of rubberlike materials,” Section 22.5.1). In addition, this model cannot be used with elements based on the plane stress assumption (shell, membrane, and continuum plane stress elements). Hybrid elements can be used with this model only when the accompanying hyperelasticity definition is completely incompressible. When this model is used with reduced-integration elements, the instantaneous elastic moduli are used to calculate the default hourglass stiffness.

Output

In addition to the standard output identifiers available in Abaqus/Standard (“Abaqus/Standard output variable identifiers,” Section 4.2.1), the following variables have special meaning if hysteretic behavior is defined:

EE	Elastic strain corresponding to the stress state at time t and the instantaneous elastic material properties.
CE	Equivalent creep strain defined as the difference between the total strain and the elastic strain.

These strain measures are used to approximate the strain energy, SENER, and the viscous dissipation, CENER. These approximations may lead to underestimation of the strain energy and overestimation of the viscous dissipation since the effects of internal stresses on these energy quantities are neglected. This inaccuracies may be particularly noticeable in the case of nonmonotonic loading.

Additional references

- Bergstrom, J. S., and M. C. Boyce, “Constitutive Modeling of the Large Strain Time-Dependent Behavior of Elastomers,” Journal of the Mechanics and Physics of Solids, vol. 46, no. 5, pp. 931–954, May 1998.
- Bergstrom, J. S., and M. C. Boyce, “Constitutive Modeling of the Time-Dependent and Cyclic Loading of Elastomers and Application to Soft Biological Tissues,” Mechanics of Materials, vol. 33, no. 9, pp. 523–530, 2001.

22.8.2 PARALLEL NETWORK VISCOELASTIC MODEL

Product: Abaqus/Standard

References

- "Material library: overview," Section 21.1.1
- "Combining material behaviors," Section 21.1.3
- "Inelastic behavior," Section 23.1.1
- *HYPERELASTIC
- *VISCOELASTIC

Overview

The parallel network nonlinear viscoelastic model:

- is intended for modeling materials that exhibit nonlinear viscous behavior and undergo large deformations;
- consists of multiple elastic and viscoelastic networks in parallel;
- uses a hyperelastic material model to specify the elastic response; and
- uses multiplicative split of the deformation gradient and a flow rule derived from a creep potential to specify the viscous behavior.

Material behavior

The parallel network nonlinear viscoelastic model consists of multiple elastic and viscoelastic networks connected in parallel, as shown in Figure 22.8.2–1. The number of viscoelastic networks, N, can be arbitrary; however, at most one purely elastic equilibrium network (network 0 in Figure 22.8.2–1) is allowed in the model. If the elastic network is not defined, the stress in the material will relax completely over time.

The model can be used to predict complex behavior of viscous materials subjected to finite strains, which cannot be modeled accurately using the linear viscoelastic model available in Abaqus (see "Time domain viscoelasticity," Section 22.7.1). An example of such complex behavior is depicted in Figure 22.8.2–2, which shows normalized stress relaxation curves for three different strain levels. This behavior can be modeled accurately using the nonlinear viscoelastic model, but it cannot be captured with the linear model. In the latter case, the three curves would coincide.

Elastic behavior

The elastic part of the response for all the networks is specified using the hyperelastic material model. Any of the hyperelastic models available in Abaqus can be used (see "Hyperelastic behavior of rubberlike materials," Section 22.5.1). The same hyperelastic material definition is used for all the networks, scaled

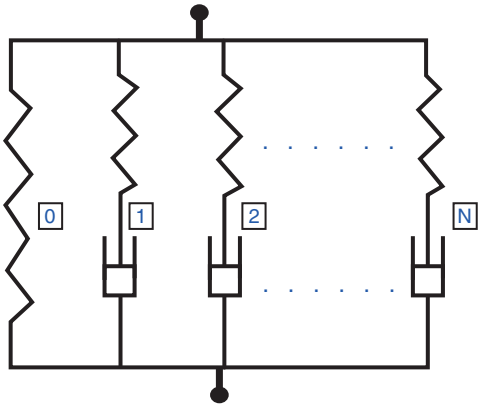


Figure 22.8.2–1 Nonlinear viscoelastic model with multiple parallel networks.

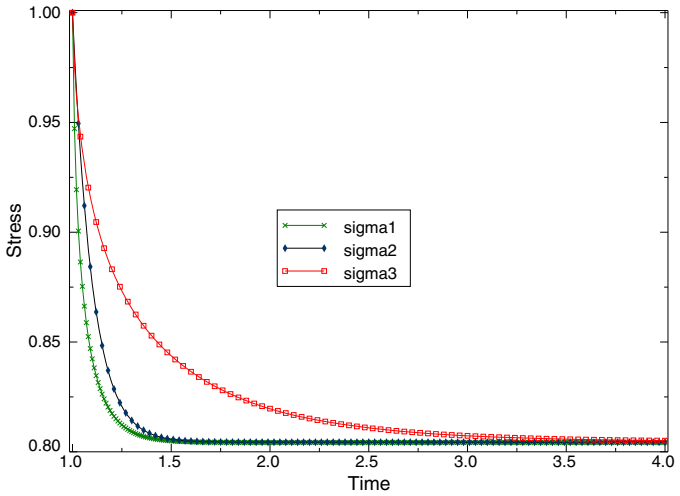


Figure 22.8.2–2 Normalized stress relaxation curves for three different strain levels.

by a stiffness ratio specific to each network. Consequently, only one hyperelastic material definition is required by the model along with the stiffness ratio for each network. The elastic response can be specified by defining either the instantaneous response or the long-term response.

Viscous behavior

Viscous behavior must be defined for each viscoelastic network. It is modeled by assuming the multiplicative split of the deformation gradient and the existence of the creep potential, G^{cr}, from which the flow rule is derived. In the multiplicative split the deformation gradient is expressed as

$$\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^{cr},$$

where $\mathbf{F}^e$ is the elastic part of the deformation gradient (representing the hyperelastic behavior) and $\mathbf{F}^{cr}$ is the creep part of the deformation gradient (representing the stress-free intermediate configuration). The creep potential is assumed to have the general form

$$G^{cr} = G^{cr}(\boldsymbol{\sigma}),$$

where $\boldsymbol{\sigma}$ is the Cauchy stress. If the potential is specified, the flow rule can be obtained from

$$\mathbf{D}^{cr} = \dot{\lambda}\frac{\partial G^{cr}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}},$$

where $\mathbf{D}^{cr}$ is the symmetric part of the velocity gradient, $\mathbf{L}^{cr}$, expressed in the current configuration and $\dot{\lambda}$ is the proportionality factor. In this model the creep potential is given by

$$G^{cr} = \bar{q},$$

and the proportionality factor is taken as $\dot{\lambda} = \dot{\bar{\varepsilon}}^{cr}$, where $\bar{q}$ is the equivalent deviatoric Cauchy stress and $\dot{\bar{\varepsilon}}^{cr}$ is the equivalent creep stain rate. In this case the flow rule has the form

$$\mathbf{D}^{cr} = \frac{3}{2\bar{q}}\dot{\bar{\varepsilon}}^{cr}\bar{\boldsymbol{\sigma}},$$

or, equivalently

$$\mathbf{D}^{cr} = \frac{3}{2\tilde{q}}\dot{\bar{\varepsilon}}^{cr}\bar{\boldsymbol{\tau}},$$

where $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ is the Kirchhoff stress, J is the determinant of $\mathbf{F}$, $\bar{\boldsymbol{\sigma}}$ is the deviatoric Cauchy stress, $\bar{\boldsymbol{\tau}}$ is the deviatoric Kirchhoff stress, and $\tilde{q} = J\bar{q}$. To complete the derivation, the evolution law for $\dot{\bar{\varepsilon}}^{cr}$ must be provided. In this model $\dot{\bar{\varepsilon}}^{cr}$ can be determined from either a power-law strain hardening model or a hyperborlic-sine model.

Power-law strain hardening model

The power-law strain hardening model is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = \left(A\tilde{q}^n[(m+1)\bar{\varepsilon}^{cr}]^m\right)^{\frac{1}{m+1}},$$

$\dot{\bar{\varepsilon}}^{cr}$	is the equivalent creep strain rate,
$\bar{\varepsilon}^{cr}$	is the equivalent creep strain,
$\tilde{q}$	is the equivalent deviatoric Kirchhoff stress, and
A, m, and n	are material parameters.

Hyperbolic-sine law model

The hyperbolic-sine law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A(\sinh B\tilde{q})^n,$$

where

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$	are defined above, and
A, B, and n	are material parameters.

Thermal expansion

Only the isotropic thermal expansion is permitted with the nonlinear viscoelastic material ("Thermal expansion," Section 26.1.2).

Defining viscoelastic response

The nonlinear viscoelastic response is defined by specifying the identifier, stiffness ratio, and creep law for each viscoelastic network.

Specifying network identifier

Each viscoelastic network in the material model must be assigned a unique network identifier or network id. The network identifiers must be consecutive integers starting with 1. The order in which they are specified is not important.

Input File Usage: Use the following option to specify the network identifier:

∗VISCOELASTIC, NONLINEAR, NETWORKID=*networkId*

Defining the stiffness ratio

The contribution of each network to the overall response of the material is determined by the value of the stiffness ratio, r, which is used to scale the elastic response of the network material. The sum of the stiffness ratios of the viscoelastic networks must be smaller than or equal to 1. If the sum of the ratios is equal to 1, the purely elastic equilibrium network is not created. If the sum of the ratios is smaller than 1, the equilibrium network is created with a stiffness ratio, r_0, equal to

$$r_0 = 1 - \sum_{k=1}^{N} r_k,$$

where N denotes the number of viscoelastic networks and r_k is the stiffness ratio of network k.

Input File Usage: Use the following option to specify the network's stiffness ratio:

∗VISCOELASTIC, NONLINEAR, SRATIO=*ratio*

Specifying the creep law

The definition of creep behavior in Abaqus/Standard is completed by specifying the creep law.

Strain hardening power law creep model

The strain hardening law is defined by specifying three material parameters: A, n, and m. For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$.

Input File Usage: *VISCOELASTIC, NONLINEAR, LAW=STRAIN

Hyperbolic sine creep model

The hyperbolic sine creep law is specified by providing three nonnegative parameters: A, B, and n.

Input File Usage: *VISCOELASTIC, NONLINEAR, LAW=HYPERB

Material response in different analysis steps

The material is active during all stress/displacement procedure types. However, the creep effects are taken into account only in a quasi-static analysis (see "Quasi-static analysis," Section 6.2.5). In other stress/displacement procedures the evolution of the state variables is suppressed and the creep strain remains unchanged.

Elements

The nonlinear viscoelastic model is available with continuum elements that include mechanical behavior (elements that have displacement degrees of freedom), except for one-dimensional and plane stress elements.

Output

In addition to the standard output identifiers available in Abaqus/Standard ("Abaqus/Standard output variable identifiers," Section 4.2.1), the following variables have special meaning for the nonlinear viscoelastic material model:

CEEQ	The overall equivalent creep strain, defined as $\bar{\varepsilon}^{cr} = \sum_{k=1}^{N} r_k \bar{\varepsilon}_k^{cr}$.
CE	The overall creep strain, defined as $\boldsymbol{\varepsilon}^{cr} = \sum_{k=1}^{N} r_k \boldsymbol{\varepsilon}_k^{cr}$.
CENER	The overall viscous dissipated energy per unit volume, defined as $E_c = \sum_{k=1}^{N} E_c^k$.
SENER	The overall elastic strain energy density per unit volume, defined as $E_s = \sum_{k=0}^{N} E_s^k$.

In the above definitions r_k denotes the stiffness ratio for network k, N denotes the number of viscoelastic networks, the subscript or superscript k is used to denote network quantities, and the network 0 is assumed to be the purely elastic network.

22.9 Rate sensitive elastomeric foams

- “Low-density foams,” Section 22.9.1

22.9.1 LOW-DENSITY FOAMS

Products: Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- *LOW DENSITY FOAM
- *UNIAXIAL TEST DATA
- "Creating a low-density foam material model" in "Defining elasticity," Section 12.9.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The low-density foam material model:

- is intended for low-density, highly compressible elastomeric foams with significant rate sensitive behavior (such as polyurethane foam);
- requires the direct specification of uniaxial stress-strain curves at different strain rates for both tension and compression;
- optionally allows the specification of lateral strain data to include Poisson effects;
- allows for the specification of optional unloading stress-strain curves for better representation of the hysteretic behavior and energy absorption during cyclic loading; and
- requires that geometric nonlinearity be accounted for during the analysis step (see "Defining an analysis," Section 6.1.2, and "General and linear perturbation procedures," Section 6.1.3), since it is intended for finite-strain applications.

Mechanical response

Low-density, highly compressible elastomeric foams are widely used in the automotive industry as energy absorbing materials. Foam padding is used in many passive safety systems, such as behind headliners for head impact protection, in door trims for pelvis and thorax protection, etc. Energy absorbing foams are also commonly used in packaging of hand-held and other electronic devices.

The low-density foam material model in Abaqus/Explicit is intended to capture the highly strain-rate sensitive behavior of these materials. The model uses a pseudo visco-hyperelastic formulation whereby the strain energy potential is constructed numerically as a function of principal stretches and a set of internal variables associated with strain rate. By default the Poisson's ratio of the material is assumed to be zero. With this assumption, the evaluation of the stress-strain response becomes uncoupled along the principal deformation directions. Optionally, nonzero Poisson effects can be specified to include coupling along the principal directions.

The model requires as input the stress-strain response of the material for both uniaxial tension and uniaxial compression tests. Poisson effects can be included by also specifying lateral strain data for each of these tests. The tests can be performed at different strain rates. For each test the strain data should be given in nominal strain values (change in length per unit of original length), and the stress data should be given in nominal stress values (force per unit of original cross-sectional area). Uniaxial tension and compression curves are specified separately. The uniaxial stress and strain data are given in absolute values (positive in both tension and compression). On the other hand, when specified, the lateral strain data must be negative in tension and positive in compression, corresponding to a positive Poisson's effect. The model does not support negative Poisson's effect. Rate-dependent behavior is specified by providing the uniaxial stress-strain curves for different values of nominal strain rates.

Both loading and unloading rate-dependent curves can be specified to better characterize the hysteretic behavior and energy absorption properties of the material during cyclic loading. Use positive values of nominal strain rates for loading curves and negative values for the unloading curves. Currently this option is available only with linear strain rate regularization (see "Regularization of strain-rate-dependent data in Abaqus/Explicit" in "Material data definition," Section 21.1.2). When the unloading behavior is not specified directly, the model assumes that unloading occurs along the loading curve associated with the smallest deformation rate. A representative schematic of typical rate-dependent uniaxial compression data is shown in Figure 22.9.1–1 with both loading and unloading curves. It is important that the specified rate-dependent stress-strain curves do not intersect. Otherwise, the material is unstable, and Abaqus issues an error message if an intersection between curves is found.

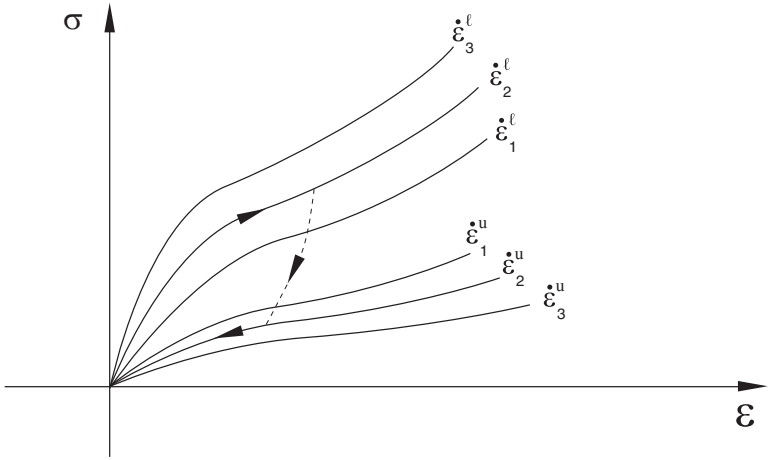


Figure 22.9.1–1 Rate-dependent loading/unloading stress-strain curves.

During the analysis, the stress along each principal deformation direction is evaluated by interpolating the specified loading/unloading stress-strain curves using the corresponding values of principal nominal strain and strain rate. The stress is then corrected by a coupling term if non-zero

Poisson effects are included. The representative response of the model for a uniaxial compression cycle is shown in Figure 22.9.1–1.

Input File Usage: Use the following options to specify a low-density foam material:

```
*LOW DENSITY FOAM
*UNIAXIAL TEST DATA, DIRECTION=TENSION
*UNIAXIAL TEST DATA, DIRECTION=COMPRESSION
```

Input File Usage: Use the first option to specify a low-density foam material with zero Poisson's ratio (default), or use the second option to include Poisson effects by defining lateral strains as part of the test data input:

```
*LOW DENSITY FOAM,LATERAL STRAIN DATA=NO (default)
*LOW DENSITY FOAM, LATERAL STRAIN DATA=YES
```

In addition, use these two options to give the experimental stress-strain data

```
*UNIAXIAL TEST DATA, DIRECTION=TENSION
*UNIAXIAL TEST DATA, DIRECTION=COMPRESSION
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: **Uniaxial Test Data→Uniaxial Tension Test Data**, **Uniaxial Test Data→Uniaxial Compression Test Data**

Relaxation coefficients

Unphysical jumps in stress due to sudden changes in the deformation rate are prevented using a technique based on viscous regularization. This technique also models stress relaxation effects in a very simplistic manner. In the case of a uniaxial test, for example, the relaxation time is given as $\tau = \mu_0 + \mu_1|\lambda - 1|^\alpha$, where μ_0, μ_1, and α are material parameters and λ is the stretch. μ_0 is a linear viscosity parameter that controls the relaxation time when $\lambda \approx 1$, and typically small values of this parameter should be used. μ_1 is a nonlinear viscosity parameter that controls the relaxation time at higher values of deformation. The smaller this value, the shorter the relaxation time. α controls the sensitivity of the relaxation speed to the stretch. The default values of these parameters are $\mu_0 = 0.0001$ (time units), $\mu_1 = 0.005$ (time units), and $\alpha = 2$.

Input File Usage: Use the following option to specify relaxation coefficients:

```
*LOW DENSITY FOAM
μ0, μ1, α
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: **Relaxation coefficients:** *mu0*, *mu1*, *alpha*

Strain rate

When Poisson's ratio is zero, three different strain rate measures can be used for the evaluation of the stress-strain response along each principal deformation direction for general three-dimensional deformation states: the nominal volumetric strain rate, the nominal strain rate along each principal

deformation direction, or the maximum of the nominal strain rates along the principal deformation directions. By default, the nominal volumetric strain rate is used; this approach does not produce rate-sensitive behavior under volume-preserving deformation modes (e.g., simple shear). Alternatively, each principal stress can be evaluated based either on the nominal strain rate along the corresponding principal direction or the maximum of the nominal strain rates; both these approaches can provide rate-sensitive behavior for volume-preserving deformation modes. All three strain rate measures produce identical rate-dependent behavior for uniaxial loading conditions when the Poisson's ratio is zero.

When non-zero Poisson effects are defined, the model uses the maximum nominal strain rate along the principal deformation directions for the evaluation of the stress-strain response. This is the default and only strain rate measure available for this case.

Input File Usage: Use the following option to use the volumetric strain rate (default when Poisson's ratio is zero):

*LOW DENSITY FOAM, STRAIN RATE=VOLUMETRIC

Use the following option to use the nominal strain rate evaluated along each principal direction:

*LOW DENSITY FOAM, STRAIN RATE=PRINCIPAL

Use the following option to use the maximum of the nominal strain rates along the principal directions (default and only option available when Poisson's ratio is not zero):

*LOW DENSITY FOAM, STRAIN RATE=MAX PRINCIPAL

Abaqus/CAE Usage: Use the following option to use the volumetric strain rate (default):

Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: **Strain rate measure: Volumetric**

Use the following option to use the strain rate evaluated along each principal direction:

Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: **Strain rate measure: Principal**

Extrapolation of stress-strain curves

By default, for this material model and for strain values beyond the range of specified strains, Abaqus/Explicit extrapolates the stress-strain curves using the slope at the last data point.

When the strain rate value exceeds the maximum specified strain rate, Abaqus/Explicit uses the stress-strain curve corresponding to the maximum specified strain rate by default. You can override this default and activate strain rate extrapolation based on the slope (with respect to strain rate).

Input File Usage: Use the following option to activate strain rate extrapolation of loading curves:

*LOW DENSITY FOAM, RATE EXTRAPOLATION=YES

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: toggle on **Extrapolate stress-strain curve beyond maximum strain rate**

Tension cutoff and failure

Low-density foams have limited strength in tension and can easily rupture under excessive tensile loading. The model in Abaqus/Explicit provides the option to specify a cutoff value for the maximum principal tensile stress that the material can sustain. The maximum principal stresses computed by the program will stay at or below this value. You can also activate deletion (removal) of the element from the simulation when the tension cutoff value is reached, which provides a simple method for modeling rupture.

Input File Usage: Use the following option to define a tension cutoff value without element deletion:

∗LOW DENSITY FOAM, TENSION CUTOFF=*value*

Use the following option to allow element deletion when the tension cutoff value is met:

∗LOW DENSITY FOAM, TENSION CUTOFF=*value*, FAIL=YES

Abaqus/CAE Usage: Use the following option to define a tension cutoff value:

Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: toggle on **Maximum allowable principal tensile stress:** *value*

Use the following option to allow element deletion when the tension cutoff value is met:

Property module: material editor: **Mechanical→Elasticity→Low Density Foam**: toggle on **Remove elements exceeding maximum**

Thermal expansion

Only isotropic thermal expansion is permitted with the low-density foam material model.

The elastic volume ratio, $J^{e\ell}$, relates the total volume ratio (current volume/reference volume), J, and the thermal volume ratio, J^{th}, via the simple relationship:

$$J^{e\ell} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = (1 + \varepsilon^{th})^3,$$

where ε^{th} is the linear thermal expansion strain that is obtained from the temperature and the isotropic thermal expansion coefficient (“Thermal expansion,” Section 26.1.2).

Material stability

The Drucker stability condition for a compressible material requires that the change in the Kirchhoff stress, $d\boldsymbol{\tau}$, following from an infinitesimal change in the logarithmic strain, $d\boldsymbol{\varepsilon}$, satisfies the inequality

$$d\boldsymbol{\tau} : d\boldsymbol{\varepsilon} > 0,$$

where the Kirchhoff stress $\boldsymbol{\tau} = J\boldsymbol{\sigma}$. Using $d\boldsymbol{\tau} = \mathbf{D} : d\boldsymbol{\varepsilon}$, the inequality becomes

$$d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0.$$

This restriction requires that the tangential material stiffness $\mathbf{D}$ be positive definite.

For an isotropic elastic formulation the inequality can be represented in terms of the principal stresses and strains

$$d\tau_1 d\varepsilon_1 + d\tau_2 d\varepsilon_2 + d\tau_3 d\varepsilon_3 > 0.$$

Thus, the relation between changes in the stress and changes in the strain can be obtained in the form of the matrix equation

$$\begin{Bmatrix} d\tau_1 \\ d\tau_2 \\ d\tau_3 \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \\ d\varepsilon_3 \end{Bmatrix},$$

where $D_{ij} = \delta\tau_i / \delta\varepsilon_j$. Since $\mathbf{D}$ must be positive definite, it is necessary that

$$D_{11} + D_{22} + D_{33} > 0,$$

$$D_{11}D_{22} + D_{22}D_{33} + D_{33}D_{11} - D_{23}^2 - D_{13}^2 - D_{12}^2 > 0,$$

$$\det(\mathbf{D}) > 0.$$

When Poisson's ratio is zero, the off diagonal terms of $\mathbf{D}$ become zero. In that case the necessary conditions for a positive definite matrix reduce to $D_{ii} > 0$; that is, the slope of the specified uniaxial stress-strain curves in the space of Kirchhoff stress versus logarithmic strain must be positive.

You should be careful defining the input data for the low-density foam model to ensure stable material response for all strain rates. If an instability is found, Abaqus issues a warning message and prints the lowest value of strain for which the instability is observed. Ideally, no instability should occur. If instabilities are observed at strain levels that are likely to occur in the analysis, it is strongly recommended that you carefully examine and revise the material input data. When nonzero Poisson effects are defined, it is highly recommended that you provide uniaxial test data in tension and compression for the same range of strain rates.

Elements

The low-density foam model can be used with solid (continuum) elements and generalized plane strain elements. One-dimensional solid elements (truss and rebar) are also available for the case when no lateral strains are specified (Poisson's ratio is zero). The model cannot be used with shells, membranes, or the Eulerian elements (EC3D8R and EC3D8RT).

Procedures

The low-density foam model must always be used with geometrically nonlinear analyses ("General and linear perturbation procedures," Section 6.1.3).

23. Inelastic Mechanical Properties

Overview 23.1
Metal plasticity 23.2
Other plasticity models 23.3
Fabric materials 23.4
Jointed materials 23.5
Concrete 23.6
Permanent set in rubberlike materials 23.7

23.1 Overview

- “Inelastic behavior,” Section 23.1.1

23.1.1 INELASTIC BEHAVIOR

The material library in Abaqus includes several models of inelastic behavior:

- **Classical metal plasticity:** The yield and inelastic flow of a metal at relatively low temperatures, where loading is relatively monotonic and creep effects are not important, can typically be described with the classical metal plasticity models ("Classical metal plasticity," Section 23.2.1). In Abaqus these models use standard Mises or Hill yield surfaces with associated plastic flow. Perfect plasticity and isotropic hardening definitions are both available in the classical metal plasticity models. Common applications include crash analyses, metal forming, and general collapse studies; the models are simple and adequate for such cases.
- **Models for metals subjected to cyclic loading:** A linear kinematic hardening model or a nonlinear isotropic/kinematic hardening model ("Models for metals subjected to cyclic loading," Section 23.2.2) can be used in Abaqus to simulate the behavior of materials that are subjected to cyclic loading. The evolution law in these models consists of a kinematic hardening component (which describes the translation of the yield surface in stress space) and, for the nonlinear isotropic/kinematic hardening model, of an isotropic component (which describes the change of the elastic range). The Bauschinger effect and plastic shakedown can be modeled with both models, but the nonlinear isotropic/kinematic hardening model provides more accurate predictions. Ratchetting and relaxation of the mean stress are accounted for only by the nonlinear isotropic/kinematic model. In addition to these two models, the ORNL model in Abaqus/Standard can be used when simple life estimation is desired for the design of stainless steels subjected to low-cycle fatigue and creep fatigue (see below).
- **Rate-dependent yield:** As strain rates increase, many materials show an increase in their yield strength. Rate dependence ("Rate-dependent yield," Section 23.2.3) can be defined in Abaqus for a number of plasticity models. Rate dependence can be used in both static and dynamic procedures. Applicable models include classical metal plasticity, extended Drucker-Prager plasticity, and crushable foam plasticity.
- **Creep and swelling:** Abaqus/Standard provides a material model for classical metal creep behavior and time-dependent volumetric swelling behavior ("Rate-dependent plasticity: creep and swelling," Section 23.2.4). This model is intended for relatively slow (quasi-static) inelastic deformation of a model such as the high-temperature creeping flow of a metal or a piece of glass. The creep strain rate is assumed to be purely deviatoric, meaning that there is no volume change associated with this part of the inelastic straining. Creep can be used with the classical metal plasticity model, with the ORNL model, and to define rate-dependent gasket behavior ("Defining the gasket behavior directly using a gasket behavior model," Section 32.6.6). Swelling can be used with the classical metal plasticity model. (Usage with the Drucker-Prager models is explained below.)
- **Annealing or melting:** Abaqus provides a modeling capability for situations in which a loss of memory related to hardening occurs above a certain user-defined temperature, known as the annealing temperature ("Annealing or melting," Section 23.2.5). It is intended for use with metals subjected to high-temperature deformation processes, in which the material may undergo melting and possibly resolidification or some other form of annealing. In Abaqus annealing or melting can be modeled

with classical metal plasticity (Mises and Hill); in Abaqus/Explicit annealing or melting can also be modeled with Johnson-Cook plasticity. The annealing temperature is assumed to be a material property. See "Annealing procedure," Section 6.12.1, for information on an alternative method for simulating annealing in Abaqus/Explicit.

- **Anisotropic yield and creep:** Abaqus provides an anisotropic yield model ("Anisotropic yield/creep," Section 23.2.6), which is available for use with materials modeled with classical metal plasticity ("Classical metal plasticity," Section 23.2.1), kinematic hardening ("Models for metals subjected to cyclic loading," Section 23.2.2), and/or creep ("Rate-dependent plasticity: creep and swelling," Section 23.2.4) that exhibit different yield stresses in different directions. The Abaqus/Standard model includes creep; creep behavior is not available in Abaqus/Explicit. The model allows for the specification of different stress ratios for each stress component to define the initial anisotropy. The model is not adequate for cases in which the anisotropy changes significantly as the material deforms as a result of loading.
- **Johnson-Cook plasticity:** The Johnson-Cook plasticity model in Abaqus/Explicit ("Johnson-Cook plasticity," Section 23.2.7) is particularly suited to model high-strain-rate deformation of metals. This model is a particular type of Mises plasticity that includes analytical forms of the hardening law and rate dependence. It is generally used in adiabatic transient dynamic analysis.
- **Dynamic failure models:** Two types of dynamic failure models are offered in Abaqus/Explicit for the Mises and Johnson-Cook plasticity models ("Dynamic failure models," Section 23.2.8). One is the shear failure model, where the failure criterion is based on the accumulated equivalent plastic strain. Another is the tensile failure model, which uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff. Both models offer a number of failure choices including element removal and are applicable mainly in truly dynamic situations. In contrast, the progressive failure and damage models (Chapter 24, "Progressive Damage and Failure") are suitable for both quasi-static and dynamic situations and have other significant advantages.
- **Porous metal plasticity:** The porous metal plasticity model ("Porous metal plasticity," Section 23.2.9) is used to model materials that exhibit damage in the form of void initiation and growth, and it can also be used for some powder metal process simulations at high relative densities (relative density is defined as the ratio of the volume of solid material to the total volume of the material). The model is based on Gurson's porous metal plasticity theory with void nucleation and is intended for use with materials that have a relative density that is greater than 0.9. The model is adequate for relatively monotonic loading.
- **Cast iron plasticity:** The cast iron plasticity model ("Cast iron plasticity," Section 23.2.10) is used to model gray cast iron, which exhibits markedly different inelastic behavior in tension and compression. The microstructure of gray cast iron consists of a distribution of graphite flakes in a steel matrix. In tension the graphite flakes act as stress concentrators, while in compression the flakes serve to transmit stresses. The resulting material is brittle in tension, but in compression it is similar in behavior to steel. The differences in tensile and compressive plastic response include: (i) a yield stress in tension that is three to five times lower than the yield stress in compression; (ii) permanent volume increase in tension, but negligible inelastic volume change in compression; (iii) different hardening behavior in tension and compression. The model is adequate for relatively monotonic loading.

- **Two-layer viscoplasticity:** The two-layer viscoplasticity model in Abaqus/Standard ("Two-layer viscoplasticity," Section 23.2.11) is useful for modeling materials in which significant time-dependent behavior as well as plasticity is observed. For metals this typically occurs at elevated temperatures. The model has been shown to provide good results for thermomechanical loading.
- **ORNL constitutive model:** The ORNL plasticity model in Abaqus/Standard ("ORNL – Oak Ridge National Laboratory constitutive model," Section 23.2.12) is intended for cyclic loading and high-temperature creep of type 304 and 316 stainless steel. Plasticity and creep calculations are provided according to the specification in Nuclear Standard NEF 9-5T, "Guidelines and Procedures for Design of Class I Elevated Temperature Nuclear System Components." This model is an extension of the linear kinematic hardening model (discussed above), which attempts to provide for simple life estimation for design purposes when low-cycle fatigue and creep fatigue are critical issues.
- **Deformation plasticity:** Abaqus/Standard provides a deformation theory Ramberg-Osgood plasticity model ("Deformation plasticity," Section 23.2.13) for use in developing fully plastic solutions for fracture mechanics applications in ductile metals. The model is most commonly applied in static loading with small-displacement analysis for which the fully plastic solution must be developed in a part of the model.
- **Extended Drucker-Prager plasticity and creep:** The extended Drucker-Prager family of plasticity models ("Extended Drucker-Prager models," Section 23.3.1) describes the behavior of granular materials or polymers in which the yield behavior depends on the equivalent pressure stress. The inelastic deformation may sometimes be associated with frictional mechanisms such as sliding of particles across each other.

  This class of models provides a choice of three different yield criteria. The differences in criteria are based on the shape of the yield surface in the meridional plane, which can be a linear form, a hyperbolic form, or a general exponent form. Inelastic time-dependent (creep) behavior coupled with the plastic behavior is also available in Abaqus/Standard for the linear form of the model. Creep behavior is not available in Abaqus/Explicit.
- **Modified Drucker-Prager/Cap plasticity and creep:** The modified Drucker-Prager/Cap plasticity model ("Modified Drucker-Prager/Cap model," Section 23.3.2) can be used to simulate geological materials that exhibit pressure-dependent yield. The addition of a cap yield surface helps control volume dilatancy when the material yields in shear and provides an inelastic hardening mechanism to represent plastic compaction. In Abaqus/Standard inelastic time-dependent (creep) behavior coupled with the plastic behavior is also available for this model; two creep mechanisms are possible: a cohesion, Drucker-Prager-like mechanism and a consolidation, cap-like mechanism.
- **Mohr-Coulomb plasticity:** The Mohr-Coulomb plasticity model ("Mohr-Coulomb plasticity," Section 23.3.3) can be used for design applications in the geotechnical engineering area. The model uses the classical Mohr-Coloumb yield criterion: a straight line in the meridional plane and an irregular hexagonal section in the deviatoric plane. However, the Abaqus Mohr-Coulomb model has a completely smooth flow potential instead of the classical hexagonal pyramid: the flow potential is a hyperbola in the meridional plane, and it uses the smooth deviatoric section proposed by Menétrey and Willam.
- **Critical state (clay) plasticity:** The clay plasticity model ("Critical state (clay) plasticity model," Section 23.3.4) describes the inelastic response of cohesionless soils. The model provides a reasonable match to the experimentally observed behavior of saturated clays. This model defines the inelastic

behavior of a material by a yield function that depends on the three stress invariants, an associated flow assumption to define the plastic strain rate, and a strain hardening theory that changes the size of the yield surface according to the inelastic volumetric strain.

- **Crushable foam plasticity:** The foam plasticity model ("Crushable foam plasticity models," Section 23.3.5) is intended for modeling crushable foams that are typically used as energy absorption structures; however, other crushable materials such as balsa wood can also be simulated with this model. This model is most appropriate for relatively monotonic loading. The crushable foam model with isotropic hardening is applicable to polymeric foams as well as metallic foams.
- **Jointed material:** The jointed material model in Abaqus/Standard ("Jointed material model," Section 23.5.1) is intended to provide a simple, continuum model for a material that contains a high density of parallel joint surfaces in different orientations, such as sedimentary rock. This model is intended for applications where stresses are mainly compressive, and it provides a joint opening capability when the stress normal to the joint tries to become tensile.
- **Concrete:** Three different constitutive models are offered in Abaqus for the analysis of concrete at low confining pressures: the smeared crack concrete model in Abaqus/Standard ("Concrete smeared cracking," Section 23.6.1); the brittle cracking model in Abaqus/Explicit ("Cracking model for concrete," Section 23.6.2); and the concrete damaged plasticity model in both Abaqus/Standard and Abaqus/Explicit ("Concrete damaged plasticity," Section 23.6.3). Each model is designed to provide a general capability for modeling plain and reinforced concrete (as well as other similar quasi-brittle materials) in all types of structures: beams, trusses, shells, and solids.

  The smeared crack concrete model in Abaqus/Standard is intended for applications in which the concrete is subjected to essentially monotonic straining and a material point exhibits either tensile cracking or compressive crushing. Plastic straining in compression is controlled by a "compression" yield surface. Cracking is assumed to be the most important aspect of the behavior, and the representation of cracking and postcracking anisotropic behavior dominates the modeling.

  The brittle cracking model in Abaqus/Explicit is intended for applications in which the concrete behavior is dominated by tensile cracking and compressive failure is not important. The model includes consideration of the anisotropy induced by cracking. In compression, the model assumes elastic behavior. A simple brittle failure criterion is available to allow the removal of elements from a mesh.

  The concrete damaged plasticity model in Abaqus/Standard and Abaqus/Explicit is based on the assumption of scalar (isotropic) damage and is designed for applications in which the concrete is subjected to arbitrary loading conditions, including cyclic loading. The model takes into consideration the degradation of the elastic stiffness induced by plastic straining both in tension and compression. It also accounts for stiffness recovery effects under cyclic loading.
- **Progressive damage and failure:** Abaqus/Explicit offers a general capability for modeling progressive damage and failure in ductile metals and fiber-reinforced composites (Chapter 24, "Progressive Damage and Failure").

Plasticity theories

Most materials of engineering interest initially respond elastically. Elastic behavior means that the deformation is fully recoverable: when the load is removed, the specimen returns to its original shape.

If the load exceeds some limit (the "yield load"), the deformation is no longer fully recoverable. Some part of the deformation will remain when the load is removed, as, for example, when a paperclip is bent too much or when a billet of metal is rolled or forged in a manufacturing process. Plasticity theories model the material's mechanical response as it undergoes such nonrecoverable deformation in a ductile fashion. The theories have been developed most intensively for metals, but they are also applied to soils, concrete, rock, ice, crushable foam, and so on. These materials behave in very different ways. For example, large values of pure hydrostatic pressure cause very little inelastic deformation in metals, but quite small hydrostatic pressure values may cause a significant, nonrecoverable volume change in a soil sample. Nonetheless, the fundamental concepts of plasticity theories are sufficiently general that models based on these concepts have been developed successfully for a wide range of materials.

Most of the plasticity models in Abaqus are "incremental" theories in which the mechanical strain rate is decomposed into an elastic part and a plastic (inelastic) part. Incremental plasticity models are usually formulated in terms of

- a *yield surface*, which generalizes the concept of "yield load" into a test function that can be used to determine if the material responds purely elastically at a particular state of stress, temperature, etc;
- a *flow rule*, which defines the inelastic deformation that occurs if the material point is no longer responding purely elastically; and
- evolution laws that define the *hardening*—the way in which the yield and/or flow definitions change as inelastic deformation occurs.

Abaqus/Standard also has a "deformation" plasticity model, in which the stress is defined from the total mechanical strain. This is a Ramberg-Osgood model ("Deformation plasticity," Section 23.2.13) and is intended primarily for ductile fracture mechanics applications, where fully plastic solutions are often required.

Elastic response

The Abaqus plasticity models also need an elasticity definition to deal with the recoverable part of the strain. In Abaqus the elasticity is defined by including linear elastic behavior or, if relevant for some plasticity models, porous elastic behavior in the same material definition (see "Material data definition," Section 21.1.2). In the case of the Mises and Johnson-Cook plasticity models in Abaqus/Explicit the elasticity can alternatively be defined using an equation of state with associated deviatoric behavior (see "Equation of state," Section 25.2.1).

When performing an elastic-plastic analysis at finite strains, Abaqus assumes that the plastic strains dominate the deformation and that the elastic strains are small. This restriction is imposed by the elasticity models that Abaqus uses. It is justified because most materials have a well-defined yield point that is a very small percentage of their Young's modulus; for example, the yield stress of metals is typically less than 1% of the Young's modulus of the material. Therefore, the elastic strains will also be less than this percentage, and the elastic response of the material can be modeled quite accurately as being linear.

In Abaqus/Explicit the elastic strain energy reported is updated incrementally. The incremental change in elastic strain energy (ΔE_s) is computed as $\Delta E_s = \Delta E_t - \Delta E_p$, where ΔE_t is the incremental change in total strain energy and ΔE_p is the incremental change in plastic energy dissipation. ΔE_s is much smaller than ΔE_t and ΔE_p for increments in which the deformation is almost all plastic.

Approximations in the calculations of ΔE_t and ΔE_p result in deviations from the true solutions that are insignificant compared to ΔE_t and ΔE_p but can be significant relative to ΔE_s. Typically, the elastic strain energy solution is quite accurate, but in some rare cases the approximations in the calculations of ΔE_t and ΔE_p can lead to a negative value reported for the elastic strain energy. These negative values are most likely to occur in an analysis that uses rate-dependent plasticity. As long as the absolute value of the elastic strain energy is very small compared to the total strain energy, a negative value for the elastic strain energy should not be considered an indication of a serious solution problem.

Stress and strain measures

Most materials that exhibit ductile behavior (large inelastic strains) yield at stress levels that are orders of magnitude less than the elastic modulus of the material, which implies that the relevant stress and strain measures are "true" stress (Cauchy stress) and logarithmic strain. Material data for all of these models should, therefore, be given in these measures.

If you have nominal stress-strain data for a uniaxial test and the material is isotropic, a simple conversion to true stress and logarithmic plastic strain is

$$\sigma_{\text{true}} = \sigma_{\text{nom}}(1 + \varepsilon_{\text{nom}}),$$

$$\varepsilon_{\text{ln}}^{\text{pl}} = \ln(1 + \varepsilon_{\text{nom}}) - \frac{\sigma_{\text{true}}}{E},$$

where E is the Young's modulus.

Example of stress-strain data input

The example below illustrates the input of material data for the classical metal plasticity model with isotropic hardening ("Classical metal plasticity," Section 23.2.1). Stress-strain data representing the material hardening behavior are necessary to define the model. An experimental hardening curve might appear as that shown in Figure 23.1.1–1. First yield occurs at 200 MPa (29000 lb/in^2). The material then hardens to 300 MPa (43511 lb/in^2) at one percent strain, after which it is perfectly plastic. Assuming that the Young's modulus is 200000 MPa (29×10^6 lb/in^2), the plastic strain at the one percent strain point is .01 − 300/200000=.0085. When the units are newtons and millimeters, the input is

Yield Stress	Plastic Strain
200.	0.
300.	.0085

Plastic strain values, not total strain values, are used in defining the hardening behavior. Furthermore, the first data pair must correspond with the onset of plasticity (the plastic strain value must be zero in the first pair). These concepts are applicable when hardening data are defined in a tabular form for any of the following plasticity models:

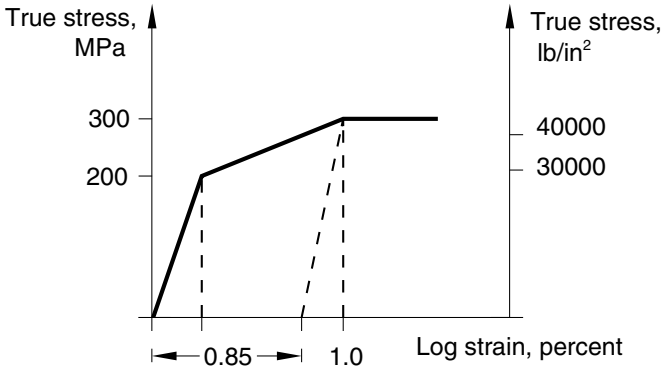


Figure 23.1.1–1 Experimental hardening curve.

- "Classical metal plasticity," Section 23.2.1
- "Models for metals subjected to cyclic loading," Section 23.2.2
- "Porous metal plasticity," Section 23.2.9 (isotropic hardening classical metal plasticity must be defined for use with this model)
- "Cast iron plasticity," Section 23.2.10
- "ORNL – Oak Ridge National Laboratory constitutive model," Section 23.2.12
- "Extended Drucker-Prager models," Section 23.3.1
- "Modified Drucker-Prager/Cap model," Section 23.3.2
- "Mohr-Coulomb plasticity," Section 23.3.3
- "Critical state (clay) plasticity model," Section 23.3.4
- "Crushable foam plasticity models," Section 23.3.5
- "Concrete smeared cracking," Section 23.6.1

The input required to define hardening is discussed in the referenced sections.

Specifying initial equivalent plastic strains

Initial values of equivalent plastic strain can be specified in Abaqus for elements that use classical metal plasticity ("Classical metal plasticity," Section 23.2.1) or Drucker-Prager plasticity ("Extended Drucker-Prager models," Section 23.3.1) by defining initial hardening conditions ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). The equivalent plastic strain (output variable PEEQ) then contains the initial value of equivalent plastic strain plus any additional equivalent plastic strain due to plastic straining during the analysis. However, the plastic strain tensor (output variable PE) contains only the amount of straining due to deformation during the analysis.

The simple one-dimensional example shown in Figure 23.1.1–2 illustrates the concept. The material is in an annealed configuration at point A; its yield stress is σ_B^0. It is then hardened by loading it along the path (A, B, C, D); the new yield stress is σ_E^0. A new analysis that employs the same hardening curve

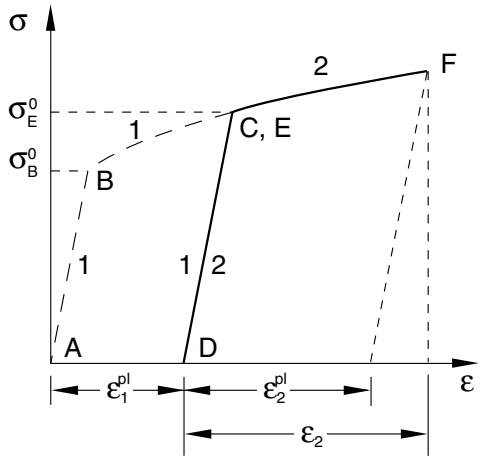


Figure 23.1.1–2 Initial equivalent plastic strain example.

as the first analysis takes this material along the path (D, E, F), starting from point ***D***, by specifying a total strain, ε_2. Plastic strain ε_2^{pl} will result and can be output (for instance) using output variable PE11. To obtain the correct yield stress, σ_E^0, the equivalent plastic strain at point ***E***, ε_1^{pl}, should be provided as an initial condition. Likewise, the correct yield stress at point ***F*** is obtained from an equivalent plastic strain PEEQ $= \varepsilon_1^{pl} + \varepsilon_2^{pl}$.

23.2 Metal plasticity

- "Classical metal plasticity," Section 23.2.1
- "Models for metals subjected to cyclic loading," Section 23.2.2
- "Rate-dependent yield," Section 23.2.3
- "Rate-dependent plasticity: creep and swelling," Section 23.2.4
- "Annealing or melting," Section 23.2.5
- "Anisotropic yield/creep," Section 23.2.6
- "Johnson-Cook plasticity," Section 23.2.7
- "Dynamic failure models," Section 23.2.8
- "Porous metal plasticity," Section 23.2.9
- "Cast iron plasticity," Section 23.2.10
- "Two-layer viscoplasticity," Section 23.2.11
- "ORNL – Oak Ridge National Laboratory constitutive model," Section 23.2.12
- "Deformation plasticity," Section 23.2.13

23.2.1 CLASSICAL METAL PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Rate-dependent yield," Section 23.2.3
- "Anisotropic yield/creep," Section 23.2.6
- "Johnson-Cook plasticity," Section 23.2.7
- Chapter 24, "Progressive Damage and Failure"
- "Dynamic failure models," Section 23.2.8
- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "UHARD," Section 1.1.35 of the Abaqus User Subroutines Reference Manual
- *PLASTIC
- *RATE DEPENDENT
- *POTENTIAL
- "Defining classical metal plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The classical metal plasticity models:

- use Mises or Hill yield surfaces with associated plastic flow, which allow for isotropic and anisotropic yield, respectively;
- use perfect plasticity or isotropic hardening behavior;
- can be used when rate-dependent effects are important;
- are intended for applications such as crash analyses, metal forming, and general collapse studies (Plasticity models that include kinematic hardening and are, therefore, more suitable for cases involving cyclic loading are also available in Abaqus: see "Models for metals subjected to cyclic loading," Section 23.2.2.);
- can be used in any procedure that uses elements with displacement degrees of freedom;
- can be used in a fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3), a fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4), or an adiabatic thermal-stress analysis ("Adiabatic analysis," Section 6.5.4) such that plastic dissipation results in the heating of a material;
- can be used in conjunction with the models of progressive damage and failure in Abaqus ("Damage and failure for ductile metals: overview," Section 24.2.1) to specify different damage initiation

criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh;

- can be used in conjunction with the shear failure model in Abaqus/Explicit to provide a simple ductile dynamic failure criterion that allows for the removal of elements from the mesh, although the progressive damage and failure methods discussed above are generally recommended instead;
- can be used in conjunction with the tensile failure model in Abaqus/Explicit to provide a tensile spall criterion offering a number of failure choices and removal of elements from the mesh; and
- must be used in conjunction with either the linear elastic material model ("Linear elastic behavior," Section 22.2.1) or the equation of state material model ("Equation of state," Section 25.2.1).

Yield surfaces

The Mises and Hill yield surfaces assume that yielding of the metal is independent of the equivalent pressure stress: this observation is confirmed experimentally for most metals (except voided metals) under positive pressure stress but may be inaccurate for metals under conditions of high triaxial tension when voids may nucleate and grow in the material. Such conditions can arise in stress fields near crack tips and in some extreme thermal loading cases such as those that might occur during welding processes. A porous metal plasticity model is provided in Abaqus for such situations. This model is described in "Porous metal plasticity," Section 23.2.9.

Mises yield surface

The Mises yield surface is used to define isotropic yielding. It is defined by giving the value of the uniaxial yield stress as a function of uniaxial equivalent plastic strain, temperature, and/or field variables. In Abaqus/Standard the yield stress can alternatively be defined in user subroutine **UHARD**.

Input File Usage: *PLASTIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**

Hill yield surface

The Hill yield surface allows anisotropic yielding to be modeled. You must specify a reference yield stress, σ^0, for the metal plasticity model and define a set of yield ratios, R_{ij}, separately. These data define the yield stress corresponding to each stress component as $R_{ij}\sigma^0$. Hill's potential function is discussed in detail in "Anisotropic yield/creep," Section 23.2.6. Yield ratios can be used to define three common forms of anisotropy associated with sheet metal forming: transverse anisotropy, planar anisotropy, and general anisotropy.

Input File Usage: Use both of the following options:

*PLASTIC *(to specify the reference yield stress σ^0)*
*POTENTIAL *(to specify the yield ratios R_{ij})*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Potential**

Hardening

In Abaqus a perfectly plastic material (with no hardening) can be defined, or work hardening can be specified. Isotropic hardening, including Johnson-Cook hardening, is available in both Abaqus/Standard and Abaqus/Explicit. In addition, Abaqus provides kinematic hardening for materials subjected to cyclic loading.

Perfect plasticity

Perfect plasticity means that the yield stress does not change with plastic strain. It can be defined in tabular form for a range of temperatures and/or field variables; a single yield stress value per temperature and/or field variable specifies the onset of yield.

Input File Usage: *PLASTIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**

Isotropic hardening

Isotropic hardening means that the yield surface changes size uniformly in all directions such that the yield stress increases (or decreases) in all stress directions as plastic straining occurs. Abaqus provides an isotropic hardening model, which is useful for cases involving gross plastic straining or in cases where the straining at each point is essentially in the same direction in strain space throughout the analysis. Although the model is referred to as a "hardening" model, strain softening or hardening followed by softening can be defined. Isotropic hardening plasticity is discussed in more detail in "Isotropic elasto-plasticity," Section 4.3.2 of the Abaqus Theory Manual.

If isotropic hardening is defined, the yield stress, σ^0, can be given as a tabular function of plastic strain and, if required, of temperature and/or other predefined field variables. The yield stress at a given state is simply interpolated from this table of data, and it remains constant for plastic strains exceeding the last value given as tabular data.

Abaqus/Explicit will regularize the data into tables that are defined in terms of even intervals of the independent variables. In some cases where the yield stress is defined at uneven intervals of the independent variable (plastic strain) and the range of the independent variable is large compared to the smallest interval, Abaqus/Explicit may fail to obtain an accurate regularization of your data in a reasonable number of intervals. In this case the program will stop after all data are processed with an error message that you must redefine the material data. See "Material data definition," Section 21.1.2, for a more detailed discussion of data regularization.

Input File Usage: *PLASTIC, HARDENING=ISOTROPIC (default if parameter is omitted)

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Isotropic**

Johnson-Cook isotropic hardening

Johnson-Cook hardening is a particular type of isotropic hardening where the yield stress is given as an analytical function of equivalent plastic strain, strain rate, and temperature. This hardening law is suited

for modeling high-rate deformation of many materials including most metals. Hill's potential function (see "Anisotropic yield/creep," Section 23.2.6) cannot be used with Johnson-Cook hardening. For more details, see "Johnson-Cook plasticity," Section 23.2.7.

Input File Usage: *PLASTIC, HARDENING=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Johnson-Cook**

User subroutine

In Abaqus/Standard the yield stress for isotropic hardening, σ^0, can alternatively be described through user subroutine **UHARD**.

Input File Usage: *PLASTIC, HARDENING=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: User**

Kinematic hardening

Two kinematic hardening models are provided in Abaqus to model the cyclic loading of metals. The linear kinematic model approximates the hardening behavior with a constant rate of hardening. The more general nonlinear isotropic/kinematic model will give better predictions but requires more detailed calibration. For more details, see "Models for metals subjected to cyclic loading," Section 23.2.2.

Input File Usage: Use the following option to specify the linear kinematic model:

*PLASTIC, HARDENING=KINEMATIC

Use the following option to specify the nonlinear combined isotropic/kinematic model:

*PLASTIC, HARDENING=COMBINED

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Kinematic** or **Combined**

Flow rule

Abaqus uses associated plastic flow. Therefore, as the material yields, the inelastic deformation rate is in the direction of the normal to the yield surface (the plastic deformation is volume invariant). This assumption is generally acceptable for most calculations with metals; the most obvious case where it is not appropriate is the detailed study of the localization of plastic flow in sheets of metal as the sheet develops texture and eventually tears apart. So long as the details of such effects are not of interest (or can be inferred from less detailed criteria, such as reaching a forming limit that is defined in terms of strain), the associated flow models in Abaqus used with the smooth Mises or Hill yield surfaces generally predict the behavior accurately.

Rate dependence

As strain rates increase, many materials show an increase in their yield strength. This effect becomes important in many metals when the strain rates range between 0.1 and 1 per second; and it can be very important for strain rates ranging between 10 and 100 per second, which are characteristic of high-energy dynamic events or manufacturing processes.

There are multiple ways to introduce a strain-rate-dependent yield stress.

Direct tabular data

Test data can be provided as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates ($\dot{\bar{\varepsilon}}^{pl}$); one table per strain rate. Direct tabular data cannot be used with Johnson-Cook hardening. The guidelines that govern the entry of this data are provided in "Rate-dependent yield," Section 23.2.3.

Input File Usage: *PLASTIC, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Use strain-rate-dependent data**

Yield stress ratios

Alternatively, you can specify the strain rate dependence by means of a scaling function. In this case you enter only one hardening curve, the static hardening curve, and then express the rate-dependent hardening curves in terms of the static relation; that is, we assume that

$$\bar{\sigma}(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}) = \sigma^0(\bar{\varepsilon}^{pl})R(\dot{\bar{\varepsilon}}^{pl}),$$

where σ^0 is the static yield stress, $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain, $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate, and R is a ratio, defined as $R = 1.0$ at $\dot{\bar{\varepsilon}}^{pl} = 0.0$. This method is described further in "Rate-dependent yield," Section 23.2.3.

Input File Usage: Use both of the following options:

*PLASTIC *(to specify the static yield stress $\sigma^0(\bar{\varepsilon}^{pl})$)*
*RATE DEPENDENT *(to specify the ratio $R(\dot{\bar{\varepsilon}}^{pl})$)*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Rate Dependent**

User subroutine

In Abaqus/Standard user subroutine **UHARD** can be used to define a rate-dependent yield stress. You are provided the current equivalent plastic strain and equivalent plastic strain rate and are responsible for returning the yield stress and derivatives.

Input File Usage: *PLASTIC, HARDENING=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: User**

Progressive damage and failure

In Abaqus the metal plasticity material models can be used in conjunction with the progressive damage and failure models discussed in "Damage and failure for ductile metals: overview," Section 24.2.1. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), Müschenborn-Sonne forming limit diagram (MSFLD), and, in Abaqus/Explicit, Marciniak-Kuczynski (M-K) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations. This is a great advantage over the dynamic failure models discussed next.

Input File Usage: Use the following options:

*PLASTIC
*DAMAGE INITIATION
*DAMAGE EVOLUTION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion*: Suboptions→Damage Evolution**

Shear and tensile dynamic failure in Abaqus/Explicit

In Abaqus/Explicit the metal plasticity material models can be used in conjunction with the shear and tensile failure models ("Dynamic failure models," Section 23.2.8) that are applicable in truly dynamic situations; however, the progressive damage and failure models discussed above are generally preferred.

Shear failure

The shear failure model provides a simple failure criterion that is suitable for high-strain-rate deformation of many materials including most metals. It offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The shear failure criterion is based on the value of the equivalent plastic strain and is applicable mainly to high-strain-rate, truly dynamic problems. For more details, see "Dynamic failure models," Section 23.2.8.

Input File Usage: Use both of the following options:

*PLASTIC
*SHEAR FAILURE

Abaqus/CAE Usage: The shear failure model is not supported in Abaqus/CAE.

Tensile failure

The tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff. It offers a number of failure choices including element removal. Similarly to the

shear failure model, the tensile failure model is suitable for high-strain-rate deformation of metals and is applicable to truly dynamic problems. For more details, see "Dynamic failure models," Section 23.2.8.

Input File Usage: Use both of the following options:

*PLASTIC
*TENSILE FAILURE

Abaqus/CAE Usage: The tensile failure model is not supported in Abaqus/CAE.

Heat generation by plastic work

Abaqus optionally allows for plastic dissipation to result in the heating of a material. Heat generation is typically used in the simulation of bulk metal forming or high-speed manufacturing processes involving large amounts of inelastic strain where the heating of the material caused by its deformation is an important effect because of temperature dependence of the material properties. It is applicable only to adiabatic thermal-stress analysis ("Adiabatic analysis," Section 6.5.4), fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3), or fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4).

This effect is introduced by defining the fraction of the rate of inelastic dissipation that appears as a heat flux per volume.

Input File Usage: Use all of the following options in the same material data block:

*PLASTIC
*SPECIFIC HEAT
*DENSITY
*INELASTIC HEAT FRACTION

Abaqus/CAE Usage: Use all of the following options for the same material:

Property module: material editor:
Mechanical→Plasticity→Plastic
Thermal→Specific Heat
General→Density
Thermal→Inelastic Heat Fraction

Initial conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, initial equivalent plastic strain values can be provided to specify the yield stress corresponding to the work hardened state (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

User subroutine specification in Abaqus/Standard

For more complicated cases, initial conditions can be defined in Abaqus/Standard through user subroutine **HARDINI**.

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING, USER

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

Classical metal plasticity can be used with any elements that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variable has special meaning for the classical metal plasticity models:

PEEQ — Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}|_0 + \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\bar{\varepsilon}^{pl}|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions").

23.2.2 MODELS FOR METALS SUBJECTED TO CYCLIC LOADING

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "Anisotropic yield/creep," Section 23.2.6
- "UHARD," Section 1.1.35 of the Abaqus User Subroutines Reference Manual
- *CYCLIC HARDENING
- *PLASTIC
- *POTENTIAL
- "Defining classical metal plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The kinematic hardening models:

- are used to simulate the inelastic behavior of materials that are subjected to cyclic loading;
- include a linear kinematic hardening model and a nonlinear isotropic/kinematic hardening model;
- include a nonlinear isotropic/kinematic hardening model with multiple backstresses;
- can be used in any procedure that uses elements with displacement degrees of freedom;
- in Abaqus/Standard cannot be used in adiabatic analyses, and the nonlinear isotropic/kinematic hardening model cannot be used in coupled temperature-displacement analyses;
- can be used to model rate-dependent yield;
- can be used with creep and swelling in Abaqus/Standard; and
- require the use of the linear elasticity material model to define the elastic part of the response.

Yield surfaces

The kinematic hardening models used to model the behavior of metals subjected to cyclic loading are pressure-independent plasticity models; in other words, yielding of the metals is independent of the equivalent pressure stress. These models are suited for most metals subjected to cyclic loading conditions, except voided metals. The linear kinematic hardening model can be used with the Mises or Hill yield surface. The nonlinear isotropic/kinematic model can be used only with the Mises yield surface in Abaqus/Standard and with the Mises or Hill yield surface in Abaqus/Explicit. The pressure-independent yield surface is defined by the function

$$F = f(\boldsymbol{\sigma} - \boldsymbol{\alpha}) - \sigma^0 = 0,$$

where σ^0 is the yield stress and $f(\boldsymbol{\sigma} - \boldsymbol{\alpha})$ is the equivalent Mises stress or Hill's potential with respect to the backstress $\boldsymbol{\alpha}$. For example, the equivalent Mises stress is defined as

$$f(\boldsymbol{\sigma} - \boldsymbol{\alpha}) = \sqrt{\frac{3}{2}(\mathbf{S} - \boldsymbol{\alpha}^{dev}) : (\mathbf{S} - \boldsymbol{\alpha}^{dev})},$$

where $\mathbf{S}$ is the deviatoric stress tensor (defined as $\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$, where $\boldsymbol{\sigma}$ is the stress tensor, p is the equivalent pressure stress, and $\mathbf{I}$ is the identity tensor) and $\boldsymbol{\alpha}^{dev}$ is the deviatoric part of the backstress tensor.

Flow rule

The kinematic hardening models assume associated plastic flow:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\bar{\varepsilon}}^{pl} \frac{\partial F}{\partial \boldsymbol{\sigma}},$$

where $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the rate of plastic flow and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate. The evolution of the equivalent plastic strain is obtained from the following equivalent plastic work expression:

$$\sigma^0 \dot{\bar{\varepsilon}}^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl},$$

which yields $\dot{\bar{\varepsilon}}^{pl} = \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}$ for isotropic Mises plasticity. The assumption of associated plastic flow is acceptable for metals subjected to cyclic loading as long as microscopic details, such as localization of plastic flow occurring as a metal component ruptures due to cyclic fatigue loads, are not of interest.

Hardening

The linear kinematic hardening model has a constant hardening modulus, and the nonlinear isotropic/kinematic hardening model has both nonlinear kinematic and nonlinear isotropic hardening components.

Linear kinematic hardening model

The evolution law of this model consists of a linear kinematic hardening component that describes the translation of the yield surface in stress space through the backstress, $\boldsymbol{\alpha}$. When temperature dependence is omitted, this evolution law is the linear Ziegler hardening law

$$\dot{\boldsymbol{\alpha}} = C\frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha})\dot{\bar{\varepsilon}}^{pl},$$

where $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate and C is the kinematic hardening modulus. In this model the equivalent stress defining the size of the yield surface, σ^0, remains constant, $\sigma^0 = \sigma|_0$, where $\sigma|_0$ is the equivalent stress defining the size of the yield surface at zero plastic strain.

Nonlinear isotropic/kinematic hardening model

The evolution law of this model consists of two components: a nonlinear kinematic hardening component, which describes the translation of the yield surface in stress space through the backstress, $\boldsymbol{\alpha}$; and an isotropic hardening component, which describes the change of the equivalent stress defining the size of the yield surface, σ^0, as a function of plastic deformation.

The kinematic hardening component is defined to be an additive combination of a purely kinematic term (linear Ziegler hardening law) and a relaxation term (the *recall* term), which introduces the nonlinearity. In addition, several kinematic hardening components (backstresses) can be superposed, which may considerably improve results in some cases. When temperature and field variable dependencies are omitted, the hardening laws for each backstress are

$$\dot{\boldsymbol{\alpha}}_k = C_k \frac{1}{\sigma^0}(\boldsymbol{\sigma} - \boldsymbol{\alpha})\dot{\bar{\varepsilon}}^{pl} - \gamma_k \boldsymbol{\alpha}_k \dot{\bar{\varepsilon}}^{pl},$$

and the overall backstress is computed from the relation

$$\boldsymbol{\alpha} = \sum_{k=1}^{N} \boldsymbol{\alpha}_k,$$

where N is the number of backstresses, and C_k and γ_k are material parameters that must be calibrated from cyclic test data. C_k are the initial kinematic hardening moduli, and γ_k determine the rate at which the kinematic hardening moduli decrease with increasing plastic deformation. The kinematic hardening law can be separated into a deviatoric part and a hydrostatic part; only the deviatoric part has an effect on the material behavior. When C_k and γ_k are zero, the model reduces to an isotropic hardening model. When all γ_k equal zero, the linear Ziegler hardening law is recovered. Calibration of the material parameters is discussed in "Usage and calibration of the kinematic hardening models," below. Figure 23.2.2–1 shows an example of nonlinear kinematic hardening with three backstresses. Each of the backstresses covers a different range of strains, and the linear hardening law is retained for large strains.

The isotropic hardening behavior of the model defines the evolution of the yield surface size, σ^0, as a function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$. This evolution can be introduced by specifying σ^0 directly as a function of $\bar{\varepsilon}^{pl}$ in tabular form, by specifying σ^0 in user subroutine **UHARD** (in Abaqus/Standard only), or by using the simple exponential law

$$\sigma^0 = \sigma|_0 + Q_\infty(1 - e^{-b\,\bar{\varepsilon}^{pl}}),$$

where $\sigma|_0$ is the yield stress at zero plastic strain and Q_∞ and b are material parameters. Q_∞ is the maximum change in the size of the yield surface, and b defines the rate at which the size of the yield

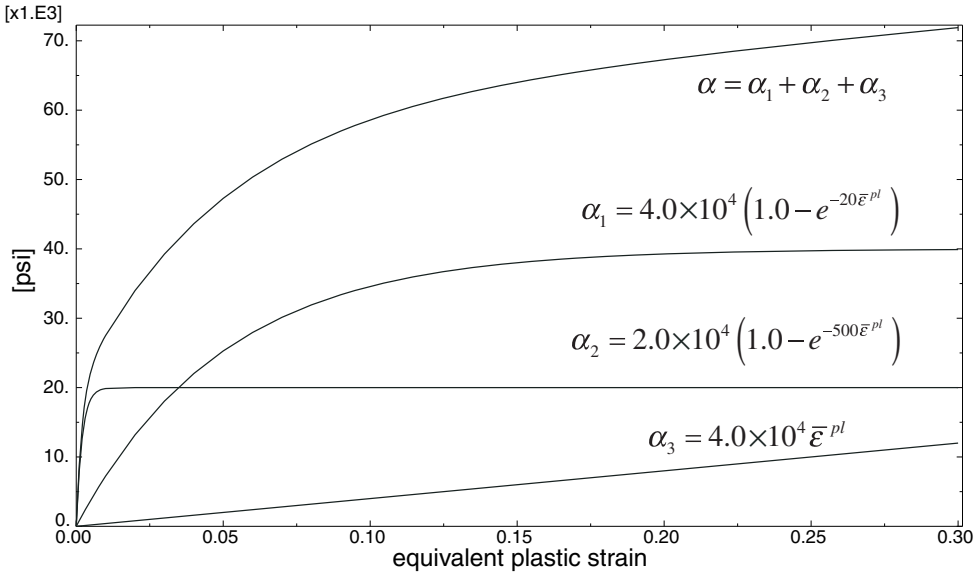


Figure 23.2.2–1 Kinematic hardening model with three backstresses.

surface changes as plastic straining develops. When the equivalent stress defining the size of the yield surface remains constant ($\sigma^0 = \sigma|_0$), the model reduces to a nonlinear kinematic hardening model.

The evolution of the kinematic and the isotropic hardening components is illustrated in Figure 23.2.2–2 for unidirectional loading and in Figure 23.2.2–3 for multiaxial loading. The evolution law for the kinematic hardening component implies that the backstress is contained within a cylinder of radius $\sqrt{2/3}\,\alpha^s = \sqrt{2/3}\,\sum_k^N C_k/\gamma_k$, where α^s is the magnitude of $\boldsymbol{\alpha}$ at saturation (large plastic strains). It also implies that any stress point must lie within a cylinder of radius $\sqrt{2/3}\,\sigma_{max}$ (using the notation of Figure 23.2.2–2) since the yield surface remains bounded. At large plastic strain any stress point is contained within a cylinder of radius $\sqrt{2/3}\,(\alpha^s + \sigma^s)$, where σ^s is the equivalent stress defining the size of the yield surface at large plastic strain. If tabular data are provided for the isotropic component, σ^s is the last value given to define the size of the yield surface. If user subroutine **UHARD** is used, this value will depend on your implementation; otherwise, $\sigma^s = \sigma|_0 + Q_\infty$.

Predicted material behavior

In the kinematic hardening models the center of the yield surface moves in stress space due to the kinematic hardening component. In addition, when the nonlinear isotropic/kinematic hardening model is used, the yield surface range may expand or contract due to the isotropic component. These features allow modeling of inelastic deformation in metals that are subjected to cycles of load or temperature, resulting in significant inelastic deformation and, possibly, low-cycle fatigue failure. These models account for the following phenomena:

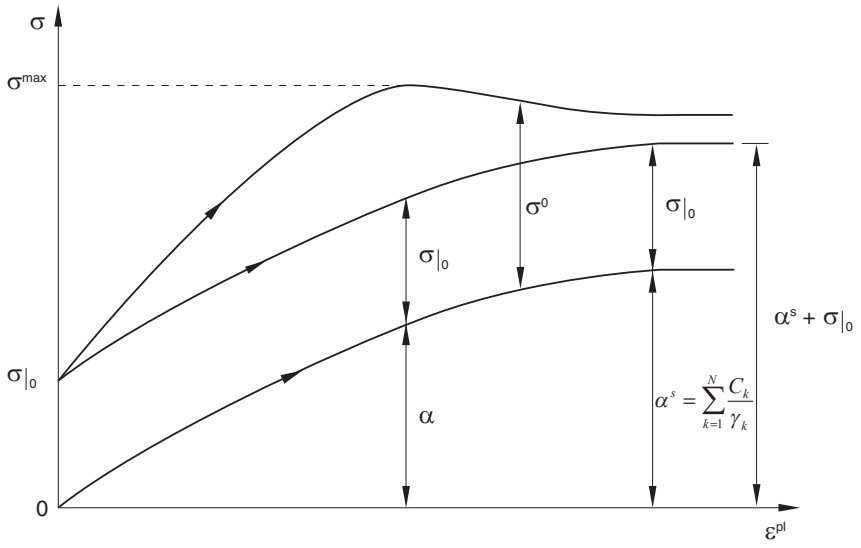


Figure 23.2.2–2 One-dimensional representation of the hardening in the nonlinear isotropic/kinematic model.

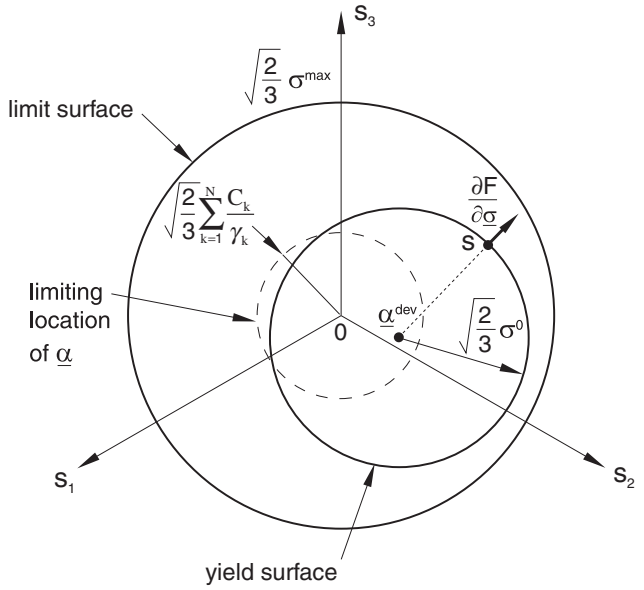


Figure 23.2.2–3 Three-dimensional representation of the hardening in the nonlinear isotropic/kinematic model.

- **Bauschinger effect:** This effect is characterized by a reduced yield stress upon load reversal after plastic deformation has occurred during the initial loading. This phenomenon decreases with continued cycling. The linear kinematic hardening component takes this effect into consideration, but a nonlinear component improves the shape of the cycles. Further improvement of the shape of the cycle can be obtained by using a nonlinear model with multiple backstresses.
- **Cyclic hardening with plastic shakedown:** This phenomenon is characteristic of symmetric stress- or strain-controlled experiments. Soft or annealed metals tend to harden toward a stable limit, and initially hardened metals tend to soften. Figure 23.2.2–4 illustrates the behavior of a metal that hardens under prescribed symmetric strain cycles.

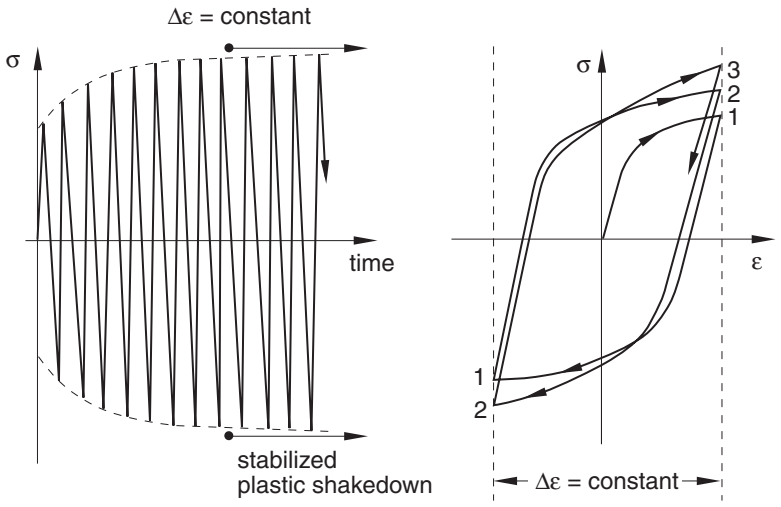


Figure 23.2.2–4 Plastic shakedown.

The kinematic hardening component of the models used alone predicts plastic shakedown after one stress cycle. The combination of the isotropic component together with the nonlinear kinematic component predicts shakedown after several cycles.

- **Ratchetting:** Unsymmetric cycles of stress between prescribed limits will cause progressive "creep" or "ratchetting" in the direction of the mean stress (Figure 23.2.2–5). Typically, transient ratchetting is followed by stabilization (zero ratchet strain) for low mean stresses, while a constant increase in the accumulated ratchet strain is observed at high mean stresses. The nonlinear kinematic hardening component, used without the isotropic hardening component, predicts constant ratchet strain. The prediction of ratchetting is improved by adding isotropic hardening, in which case the ratchet strain may decrease until it becomes constant. However, in general the nonlinear hardening model with a single backstress predicts a too significant ratchetting effect. A considerable improvement in modeling ratchetting can be achieved by superposing several kinematic hardening models (backstresses) and choosing one of the models to be linear or nearly linear ($\gamma_k \ll C_k$), which results in a less pronounced ratchetting effect.

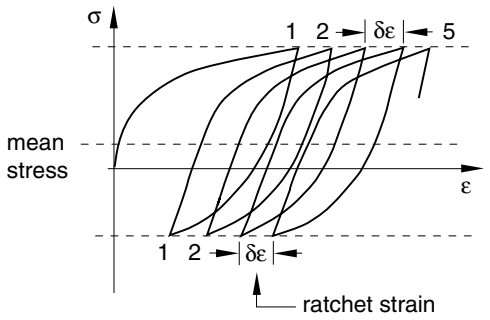


Figure 23.2.2–5 Ratchetting.

- **Relaxation of the mean stress:** This phenomenon is characteristic of an unsymmetric strain experiment, as shown in Figure 23.2.2–6.

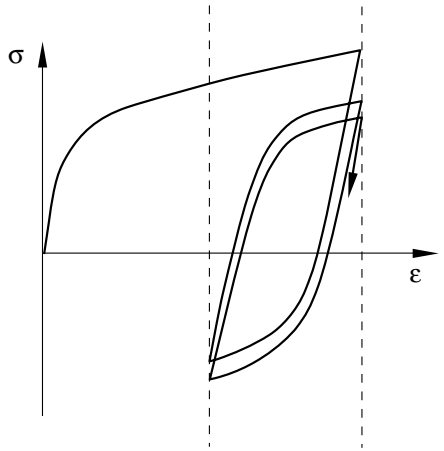


Figure 23.2.2–6 Relaxation of the mean stress.

As the number of cycles increases, the mean stress tends to zero. The nonlinear kinematic hardening component of the nonlinear isotropic/kinematic hardening model accounts for this behavior.

Limitations

The linear kinematic model is a simple model that gives only a first approximation of the behavior of metals subjected to cyclic loading, as explained above. The nonlinear isotropic/kinematic hardening model can provide more accurate results in many cases involving cyclic loading, but it still has the following limitations:

- The isotropic hardening is the same at all strain ranges. Physical observations, however, indicate that the amount of isotropic hardening depends on the magnitude of the strain range. Furthermore, if the specimen is cycled at two different strain ranges, one followed by the other, the deformation in the first cycle affects the isotropic hardening in the second cycle. Thus, the model is only a coarse approximation of actual cyclic behavior. It should be calibrated to the expected size of the strain cycles of importance in the application.
- The same cyclic hardening behavior is predicted for proportional and nonproportional load cycles. Physical observations indicate that the cyclic hardening behavior of materials subjected to nonproportional loading may be very different from uniaxial behavior at a similar strain amplitude.

The example problems "Simple proportional and nonproportional cyclic tests," Section 3.2.8 of the Abaqus Benchmarks Manual, "Notched beam under cyclic loading," Section 1.1.7 of the Abaqus Example Problems Manual and "Uniaxial ratchetting under tension and compression," Section 1.1.8 of the Abaqus Example Problems Manual, illustrate the phenomena of cyclic hardening with plastic shakedown, ratchetting, and relaxation of the mean stress for the nonlinear isotropic/kinematic hardening model, as well as its limitations.

Usage and calibration of the kinematic hardening models

The linear kinematic model approximates the hardening behavior with a constant rate of hardening. This hardening rate should be matched to the average hardening rate measured in stabilized cycles over a strain range corresponding to that expected in the application. A stabilized cycle is obtained by cycling over a fixed strain range until a steady-state condition is reached; that is, until the stress-strain curve no longer changes shape from one cycle to the next. The more general nonlinear model will give better predictions but requires more detailed calibration.

Linear kinematic hardening model

The test data obtained from a half cycle of a unidirectional tension or compression experiment must be linearized, since this simple model can predict only linear hardening. The data are usually based on measurements of the stabilized behavior in strain cycles covering a strain range corresponding to the strain range that is anticipated to occur in the application. Abaqus expects you to provide only two data pairs to define this linear behavior: the yield stress, $\sigma|_0$, at zero plastic strain and a yield stress, σ, at a finite plastic strain value, ε^{pl}. The linear kinematic hardening modulus, C, is determined from the relation

$$C = \frac{\sigma - \sigma|_0}{\varepsilon^{pl}}.$$

You can provide several sets of two data pairs as a function of temperature to define the variation of the linear kinematic hardening modulus with respect to temperature. If the Hill yield surface is desired for this model, you must specify a set of yield ratios, R_{ij}, independently (see "Anisotropic yield/creep," Section 23.2.6, for information on how to specify the yield ratios).

This model gives physically reasonable results for only relatively small strains (less than 5%).

Input File Usage: *PLASTIC, HARDENING=KINEMATIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Kinematic**

Nonlinear isotropic/kinematic hardening model

The evolution of the equivalent stress defining the size of the yield surface, σ^0, as a function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, defines the isotropic hardening component of the model. You can define this isotropic hardening component through an exponential law or directly in tabular form. It need not be defined if the yield surface remains fixed throughout the loading. In Abaqus/Explicit if the Hill yield surface is desired for this model, you must specify a set of yield ratios, R_{ij}, independently (see "Anisotropic yield/creep," Section 23.2.6, for information on how to specify the yield ratios). The Hill yield surface cannot be used with this model in Abaqus/Standard.

The material parameters C_k and γ_k determine the kinematic hardening component of the model. Abaqus offers three different ways of providing data for the kinematic hardening component of the model: the parameters C_k and γ_k can be specified directly, half-cycle test data can be given, or test data obtained from a stabilized cycle can be given. The experiments required to calibrate the model are described below.

Defining the isotropic hardening component by the exponential law

Specify the material parameters of the exponential law $\sigma|_0$, Q_∞, and b directly if they are already calibrated from test data. These parameters can be specified as functions of temperature and/or field variables.

Input File Usage: *CYCLIC HARDENING, PARAMETERS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Cyclic Hardening**: toggle on **Use parameters**.

Defining the isotropic hardening component by tabular data

Isotropic hardening can be introduced by specifying the equivalent stress defining the size of the yield surface, σ^0, as a tabular function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$. The simplest way to obtain these data is to conduct a symmetric strain-controlled cyclic experiment with strain range $\Delta\varepsilon$. Since the material's elastic modulus is large compared to its hardening modulus, this experiment can be interpreted approximately as repeated cycles over the same plastic strain range $\Delta\varepsilon^{pl} \approx \Delta\varepsilon - 2\sigma_1^t/E$ (using the notation of Figure 23.2.2–7, where E is the Young's modulus of the material). The equivalent stress defining the size of the yield surface is $\sigma|_0$ at zero equivalent plastic strain; for the peak tensile stress points it is obtained by isolating the kinematic component from the yield stress (see Figure 23.2.2–2) as

$$\sigma_i^0 = \sigma_i^t - \alpha_i$$

for each cycle *i*, where $\alpha_i = (\sigma_i^t + \sigma_i^c)/2$. Since the model predicts approximately the same backstress value in each cycle at a particular strain level, $\alpha_i \approx (\sigma_1^t + \sigma_1^c)/2$. The equivalent plastic strain corresponding to σ_i^0 is

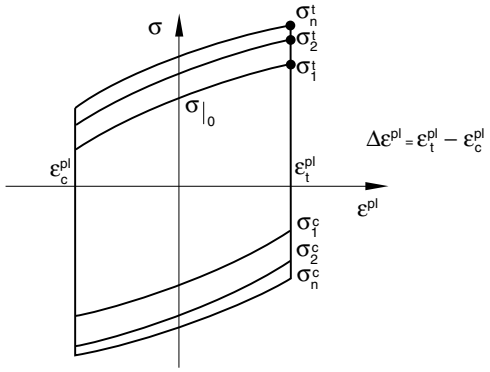


Figure 23.2.2–7 Symmetric strain cycle experiment.

$$\bar{\varepsilon}^{pl}_i = \frac{1}{2}(4i - 3)\,\Delta\varepsilon^{pl}.$$

Data pairs $(\sigma^0_i,\ \bar{\varepsilon}^{pl}_i)$, including the value $\sigma|_0$ at zero equivalent plastic strain, are specified in tabulated form. The tabulated values defining the size of the yield surface should be provided for the entire equivalent plastic strain range to which the material may be subjected. The data can be provided as functions of temperature and/or field variables.

To obtain accurate cyclic hardening data, such as would be needed for low-cycle fatigue calculations, the calibration experiment should be performed at a strain range, $\Delta\varepsilon$, that corresponds to the strain range anticipated in the analysis because the material model does not predict different isotropic hardening behavior at different strain ranges. This limitation also implies that, even though a component is made from the same material, it may have to be divided into several regions with different hardening properties corresponding to different anticipated strain ranges. Field variables and field variable dependence of these properties can also be used for this purpose.

Abaqus allows the specification of strain rate effects in the isotropic component of the nonlinear isotropic/kinematic hardening model. The rate-dependent isotropic hardening data can be defined by specifying the equivalent stress defining the size of the yield surface, σ^0, as a tabular function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, at different values of the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$.

Input File Usage: Use the following option to define isotropic hardening with tabular data:

∗CYCLIC HARDENING

Use the following option to define rate-dependent isotropic hardening with tabular data:

∗CYCLIC HARDENING, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Combined**: **Suboptions→Cyclic Hardening**

Defining the isotropic hardening component in a user subroutine in Abaqus/Standard

Specify σ^0 directly in user subroutine **UHARD**. σ^0 may be dependent on equivalent plastic strain and temperature. This method cannot be used if the kinematic hardening component is specified by using half-cycle test data.

Input File Usage: *CYCLIC HARDENING, USER

Abaqus/CAE Usage: You cannot define the isotropic hardening component in user subroutine **UHARD** in Abaqus/CAE.

Defining the kinematic hardening component by specifying the material parameters directly

The parameters C_k and γ_k can be specified directly as a function of temperature and/or field variables if they are already calibrated from test data. When γ_k depend on temperature and/or field variables, the response of the model under thermomechanical loading will generally depend on the *history* of temperature and/or field variables experienced at a material point. This dependency on temperature-history is small and fades away with increasing plastic deformation. However, if this effect is not desired, constant values for γ_k should be specified to make the material response completely independent of the history of temperature and field variables. The algorithm currently used to integrate the nonlinear isotropic/kinematic hardening model provides accurate solutions if the values of γ_k change moderately in an increment due to temperature and/or field variable dependence; however, this algorithm may not yield a solution with sufficient accuracy if the values of γ_k change abruptly in an increment.

Input File Usage: *PLASTIC, HARDENING=COMBINED, DATA TYPE=PARAMETERS, NUMBER BACKSTRESSES=n

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Combined**, **Data type: Parameters**, **Number of backstresses**: n

Defining the kinematic hardening component by specifying half-cycle test data

If limited test data are available, C_k and γ_k can be based on the stress-strain data obtained from the first half cycle of a unidirectional tension or compression experiment. An example of such test data is shown in Figure 23.2.2–8. This approach is usually adequate when the simulation will involve only a few cycles of loading.

For each data point $(\sigma_i, \varepsilon_i^{pl})$ a value of α_i (α_i is the overall backstress obtained by summing all the backstresses at this data point) is obtained from the test data as

$$\alpha_i = \sigma_i - \sigma_i^0,$$

where σ_i^0 is the user-defined size of the yield surface at the corresponding plastic strain for the isotropic hardening component or the initial yield stress if the isotropic hardening component is not defined.

Integration of the backstress evolution laws over a half cycle yields the expressions

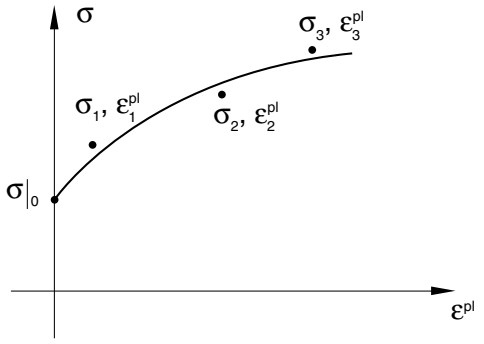


Figure 23.2.2–8 Half cycle of stress-strain data.

$$\alpha_k = \frac{C_k}{\gamma_k}(1 - e^{-\gamma_k \varepsilon^{pl}}),$$

which are used for calibrating C_k and γ_k.

When test data are given as functions of temperature and/or field variables, Abaqus determines several sets of material parameters (C_1, γ_1,..., C_N, γ_N), each corresponding to a given combination of temperature and/or field variables. Generally, this results in temperature-history (and/or field variable-history) dependent material behavior because the values of γ_k vary with changes in temperature and/or field variables. This dependency on temperature-history is small and fades away with increasing plastic deformation. However, you can make the response of the material completely independent of the history of temperature and field variables by using constant values for the parameters γ_k. This can be achieved by running a data check analysis first; an appropriate constant values of γ_k can be determined from the information provided in the data file during the data check. The values for the parameters C_k and the constant parameters γ_k can then be entered directly as described above.

If the model with multiple backstresses is used, Abaqus obtains hardening parameters for different values of initial guesses and chooses the ones that give the best correlation with the experimental data provided. However, you should carefully examine the obtained parameters. In some cases it might be advantageous to obtain hardening parameters for different numbers of backstresses before choosing the set of parameters.

Input File Usage: *PLASTIC, HARDENING=COMBINED, DATA TYPE=HALF CYCLE, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Combined**, **Data type: Half Cycle**, **Number of backstresses**: *n*

Defining the kinematic hardening component by specifying test data from a stabilized cycle

Stress-strain data can be obtained from the stabilized cycle of a specimen that is subjected to symmetric strain cycles. A stabilized cycle is obtained by cycling the specimen over a fixed strain range $\Delta\varepsilon$ until a

steady-state condition is reached; that is, until the stress-strain curve no longer changes shape from one cycle to the next. Such a stabilized cycle is shown in Figure 23.2.2–9. Each data pair $(\sigma_i, \varepsilon_i^{pl})$ must be specified with the strain axis shifted to ε_p^0, so that

$$\varepsilon_i^{pl} = \varepsilon_i - \frac{\sigma_i}{E} - \varepsilon_p^0,$$

and, thus, $\varepsilon_1^{pl} = 0$.

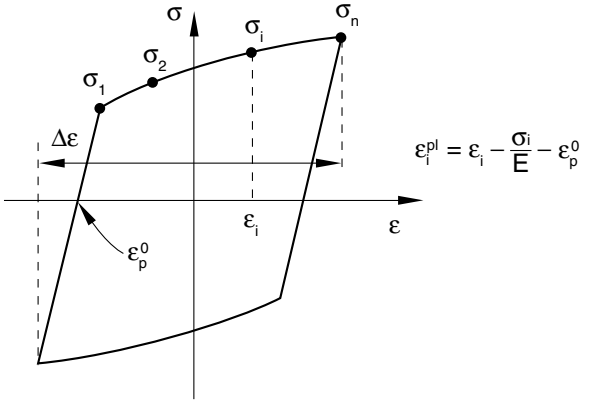


Figure 23.2.2–9 Stress-strain data for a stabilized cycle.

For each pair $(\sigma_i, \varepsilon_i^{pl})$ values of α_i (α_i is the overall backstress obtained by summing all the backstresses at this data point) are obtained from the test data as

$$\alpha_i = \sigma_i - \sigma^s,$$

where $\sigma^s = (\sigma_1 + \sigma_n)/2$ is the stabilized size of the yield surface.

Integration of the backstress evolution laws over this uniaxial strain cycle, with an exact match for the first data pair $(\sigma_1, 0)$, provides the expressions

$$\alpha_k = \frac{C_k}{\gamma_k}(1 - e^{-\gamma_k \varepsilon^{pl}}) + \alpha_{k,1}\, e^{-\gamma_k \varepsilon^{pl}},$$

where $\alpha_{k,1}$ denotes the k^{th} backstress at the first data point (initial value of the k^{th} backstress). The above equations enable calibration of the parameters C_k and γ_k.

If the shapes of the stress-strain curves are significantly different for different strain ranges, you may want to obtain several calibrated values of C_k and γ_k. The tabular data of the stress-strain curves obtained at different strain ranges can be entered directly in Abaqus. Calibrated values corresponding to each strain range are reported in the data file, together with an averaged set of parameters, if model definition data are requested (see "Controlling the amount of analysis input file processor information written to the

data file" in "Output," Section 4.1.1). Abaqus will use the averaged set in the analysis. These parameters may have to be adjusted to improve the match to the test data at the strain range anticipated in the analysis.

When test data are given as functions of temperature and/or field variables, Abaqus determines several sets of material parameters (C_1, γ_1,..., C_N, γ_N), each corresponding to a given combination of temperature and/or field variables. Generally, this results in temperature-history (and/or field variable-history) dependent material behavior because the values of γ_k vary with changes in temperature and/or field variables. This dependency on temperature-history is small and fades away with increasing plastic deformation. However, you can make the response of the material completely independent of the history of temperature and field variables by using constant values for the parameters γ_k. This can be achieved by running a data check analysis first; an appropriate constant values of γ_k can be determined from the information provided in the data file during the data check. The values for the parameters C_k and the constant parameters γ_k can then be entered directly as described above.

If the model with multiple backstresses is used, Abaqus obtains hardening parameters for different values of initial guesses and chooses the ones that give the best correlation with the experimental data provided. However, you should carefully examine the obtained parameters. In some cases it might be advantageous to obtain hardening parameters for different numbers of backstresses before choosing the set of parameters.

The isotropic hardening component should be defined by specifying the equivalent stress defining the size of the yield surface at zero plastic strain, as well as the evolution of the equivalent stress as a function of equivalent plastic strain. If this component is not defined, Abaqus will assume that no cyclic hardening occurs so that the equivalent stress defining the size of the yield surface is constant and equal to $(\sigma_1 + \sigma_n)/2$ (or the average of these quantities over several strain ranges when more than one strain range is provided). Since this size corresponds to the size of a saturated cycle, this is unlikely to provide accurate predictions of actual behavior, particularly in the initial cycles.

Input File Usage: *PLASTIC, HARDENING=COMBINED, DATA TYPE=STABILIZED, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Combined**, **Data type: Stabilized**, **Number of backstresses**: *n*

Initial conditions

When we need to study the behavior of a material that has already been subjected to some hardening, Abaqus allows you to prescribe initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, and for the backstresses, $\boldsymbol{\alpha}_k$. When the nonlinear isotropic/kinematic hardening model is used, the initial conditions for each backstress, $\boldsymbol{\alpha}_k$, must satisfy the condition

$$\sqrt{\frac{3}{2}\boldsymbol{\alpha}_k{}^{dev} : \boldsymbol{\alpha}_k{}^{dev}} \leq C_k/\gamma_k$$

for the model to produce a kinematic hardening response. Abaqus allows the specification of initial backstresses that violate these conditions. However, in this case the response corresponding to the backstress for which the condition is violated produces kinematic softening response: the magnitude

of the backstress decreases with plastic straining from its initial value to the saturation value. If the condition is violated for any of the backstresses, the overall response of the material is not guaranteed to produce kinematic hardening response. The initial condition for the backstress has no limitations when the linear kinematic hardening model is used.

You can specify the initial values of $\bar{\varepsilon}^{pl}$ and $\boldsymbol{\alpha}_k$ directly as initial conditions (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Number of backstresses**: *n*

User subroutine specification in Abaqus/Standard

For more complicated cases in Abaqus/Standard initial conditions can be defined through user subroutine **HARDINI**.

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING, USER, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**, **Number of backstresses**: *n*

Elements

These models can be used with elements in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom), except some beam elements in space. Beam elements in space that include shear stress caused by torsion (i.e., not thin-walled, open sections) and do not include hoop stress (i.e., not PIPE elements) cannot be used. In Abaqus/Explicit the kinematic hardening models can be used with any elements that include mechanical behavior, with the exception of one-dimensional elements (beams, pipes, and trusses) when the models are used with the Hill yield surface.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning for the kinematic hardening models:

ALPHA	Total kinematic hardening shift tensor components, $\alpha_{ij} (i, j \leq 3)$.
ALPHAk	k^{th} kinematic hardening shift tensor components ($1 \leq k \leq 10$).
ALPHAN	All tensor components of all the kinematic hardening shift tensors, except the total shift tensor.
PEEQ	Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\|_0 + \int_0^t \frac{\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}^{pl} dt}{\sigma^0}$, where $\bar{\varepsilon}^{pl}\|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions").

PENER — Plastic work, defined as: $W^{pl} = \int_0^t \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl} dt$,. This quantity is not guaranteed to be monotonically increasing for kinematic hardening models. To get a quantity that is monotonically increasing, the plastic dissipation needs to be computed as: $W^{pl} = \int_0^t (\boldsymbol{\sigma} - \boldsymbol{\alpha}) : \dot{\boldsymbol{\varepsilon}}^{pl} dt$. In Abaqus/Standard this quantity can be computed as a user-defined output variable in user subroutine **UVARM**.

23.2.3 RATE-DEPENDENT YIELD

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Classical metal plasticity," Section 23.2.1
- "Models for metals subjected to cyclic loading," Section 23.2.2
- "Johnson-Cook plasticity," Section 23.2.7
- "Extended Drucker-Prager models," Section 23.3.1
- "Crushable foam plasticity models," Section 23.3.5
- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *RATE DEPENDENT
- "Defining rate-dependent yield with yield stress ratios" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Rate-dependent yield:

- is needed to define a material's yield behavior accurately when the yield strength depends on the rate of straining and the anticipated strain rates are significant;
- is available only for the isotropic hardening metal plasticity models (Mises and Johnson-Cook), the isotropic component of the nonlinear isotropic/kinematic plasticity models, the extended Drucker-Prager plasticity model, and the crushable foam plasticity model;
- can be conveniently defined on the basis of work hardening parameters and field variables by providing tabular data for the isotropic hardening metal plasticity models, the isotropic component of the nonlinear isotropic/kinematic plasticity models, and the extended Drucker-Prager plasticity model;
- can be defined through specification of user-defined overstress power law parameters, yield stress ratios, or Johnson-Cook rate dependence parameters (this last option is not available for the crushable foam plasticity model and is the only option available for the Johnson-Cook plasticity model);
- cannot be used with any of the Abaqus/Standard creep models (metal creep, time-dependent volumetric swelling, Drucker-Prager creep, or cap creep) since creep behavior is already a rate-dependent mechanism; and
- in dynamic analysis should be specified such that the yield stress increases with increasing strain rate.

Work hardening dependencies

Generally, a material's yield stress, $\bar{\sigma}$ (or $\bar{B}$ for the crushable foam model), is dependent on work hardening, which for isotropic hardening models is usually represented by a suitable measure of equivalent plastic strain, $\bar{\varepsilon}^{pl}$; the inelastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and predefined field variables, f_i:

$$\bar{\sigma} = \bar{\sigma}(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i).$$

Many materials show an increase in their yield strength as strain rates increase; this effect becomes important in many metals and polymers when the strain rates range between 0.1 and 1 per second, and it can be very important for strain rates ranging between 10 and 100 per second, which are characteristic of high-energy dynamic events or manufacturing processes.

Defining hardening dependencies for various material models

Strain rate dependence can be defined by entering hardening curves at different strain rates directly or by defining yield stress ratios to specify the rate dependence independently.

Direct entry of test data

Work hardening dependencies can be given quite generally as tabular data for the isotropic hardening Mises plasticity model, the isotropic component of the nonlinear isotropic/kinematic hardening model, and the extended Drucker-Prager plasticity model. The test data are entered as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates. The yield stress must be given as a function of the equivalent plastic strain and, if required, of temperature and of other predefined field variables. In defining this dependence at finite strains, "true" (Cauchy) stress and log strain values should be used. The hardening curve at each temperature must always start at zero plastic strain. For perfect plasticity only one yield stress, with zero plastic strain, should be defined at each temperature. It is possible to define the material to be strain softening as well as strain hardening. The work hardening data are repeated as often as needed to define stress-strain curves at different strain rates. The yield stress at a given strain and strain rate is interpolated directly from these tables.

Input File Usage: Use one of the following options:

∗PLASTIC, HARDENING=ISOTROPIC, RATE=$\dot{\bar{\varepsilon}}^{pl}$
∗CYCLIC HARDENING, RATE=$\dot{\bar{\varepsilon}}^{pl}$
∗DRUCKER PRAGER HARDENING, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Use one of the following models:

Property module: material editor:
Mechanical→Plasticity→Plastic: **Hardening: Isotropic**,
Use strain-rate-dependent data
Mechanical→Plasticity→Drucker Prager: **Suboptions→Drucker Prager Hardening**: **Use strain-rate-dependent data**

Cyclic hardening is not supported in Abaqus/CAE.

Using yield stress ratios

Alternatively, and as the only means of defining rate-dependent yield stress for the Johnson-Cook and the crushable foam plasticity models, the strain rate behavior can be assumed to be separable, so that the stress-strain dependence is similar at all strain rate levels:

$$\bar{\sigma} = \sigma^0(\bar{\varepsilon}^{pl}, \theta, f_i) R(\dot{\bar{\varepsilon}}^{pl}, \theta, f_i),$$

where $\sigma^0(\bar{\varepsilon}^{pl}, \theta, f_i)$ (or $B(\bar{\varepsilon}^{pl}, \theta, f_i)$ in the foam model) is the static stress-strain behavior and $R(\dot{\bar{\varepsilon}}^{pl}, \theta, f_i)$ is the ratio of the yield stress at nonzero strain rate to the static yield stress (so that $R(0, \theta, f_i) = 1.0$).

Three methods are offered to define $\boldsymbol{R}$ in Abaqus: specifying an overstress power law, defining $\boldsymbol{R}$ directly as a tabular function, or specifying an analytical Johnson-Cook form to define $\boldsymbol{R}$.

Overstress power law

The Cowper-Symonds overstress power law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D\,(R-1)^n \quad \text{for} \quad \bar{\sigma} \geq \sigma^0 \quad (\text{or} \quad \bar{B} \geq B \quad \text{in the crushable foam model}),$$

where $D(\theta, f_i)$ and $n(\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: *RATE DEPENDENT, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Suboptions→Rate Dependent**: **Hardening: Power Law** *(available for valid plasticity models)*

Tabular function

Alternatively, $\boldsymbol{R}$ can be entered directly as a tabular function of the equivalent plastic strain rate (or the axial plastic strain rate in a uniaxial compression test for the crushable foam model), $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and field variables, f_i.

Input File Usage: *RATE DEPENDENT, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Suboptions→Rate Dependent**: **Hardening: Yield Ratio** *(available for valid plasticity models)*

Johnson-Cook rate dependence

Johnson-Cook rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \exp\left[\frac{1}{C}\,(R-1)\right] \quad \text{for} \quad \bar{\sigma} \geq \sigma^0\,,$$

where $\dot{\varepsilon}_0$ and $\boldsymbol{C}$ are material constants that do not depend on temperature and are assumed not to depend on predefined field variables. Johnson-Cook rate dependence can be used in conjunction with the Johnson-Cook plasticity model, the isotropic hardening metal plasticity models, and the extended

Drucker-Prager plasticity model (it cannot be used in conjunction with the crushable foam plasticity model).

This is the only form of rate dependence available for the Johnson-Cook plasticity model. For more details, see "Johnson-Cook plasticity," Section 23.2.7.

Input File Usage: *RATE DEPENDENT, TYPE=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Suboptions→Rate Dependent**: **Hardening: Johnson-Cook** *(available for valid plasticity models)*

Elements

Rate-dependent yield can be used with all elements that include mechanical behavior (elements that have displacement degrees of freedom).

23.2.4 RATE-DEPENDENT PLASTICITY: CREEP AND SWELLING

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "Defining the gasket behavior directly using a gasket behavior model," Section 32.6.6
- *CREEP
- *CREEP STRAIN RATE CONTROL
- *POTENTIAL
- *SWELLING
- *RATIOS
- "Defining a creep law" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual
- "Defining swelling" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The classical deviatoric metal creep behavior in Abaqus/Standard:

- can be defined using user subroutine **`CREEP`** or by providing parameters as input for some simple creep laws;
- can model either isotropic creep (using Mises stress potential) or anisotropic creep (using Hill's anisotropic stress potential);
- is active only during steps using the coupled temperature-displacement procedure, the transient soils consolidation procedure, and the quasi-static procedure;
- requires that the material's elasticity be defined as linear elastic behavior;
- can be modified to implement the auxiliary creep hardening rules specified in Nuclear Standard NEF 9-5T, "Guidelines and Procedures for Design of Class 1 Elevated Temperature Nuclear System Components"; these rules are exercised by means of a constitutive model developed by Oak Ridge National Laboratory ("ORNL – Oak Ridge National Laboratory constitutive model," Section 23.2.12);
- can be used in combination with creep strain rate control in analyses in which the creep strain rate must be kept within a certain range; and
- can potentially result in errors in calculated creep strains if anisotropic creep and plasticity occur simultaneously (discussed below).

Rate-dependent gasket behavior in Abaqus/Standard:

- uses unidirectional creep as part of the model of the gasket's thickness-direction behavior;
- can be defined using user subroutine **CREEP** or by providing parameters as input for some simple creep laws;
- is active only during steps using the quasi-static procedure; and
- requires that an elastic-plastic model be used to define the rate-independent part of the thickness-direction behavior of the gasket.

Volumetric swelling behavior in Abaqus/Standard:

- can be defined using user subroutine **CREEP** or by providing tabular input;
- can be either isotropic or anisotropic;
- is active only during steps using the coupled temperature-displacement procedure, the transient soils consolidation procedure, and the quasi-static procedure; and
- requires that the material's elasticity be defined as linear elastic behavior.

Creep behavior

Creep behavior is specified by the equivalent uniaxial behavior—the creep "law." In practical cases creep laws are typically of very complex form to fit experimental data; therefore, the laws are defined with user subroutine **CREEP**, as discussed below. Alternatively, two common creep laws are provided in Abaqus/Standard: the power law and the hyperbolic-sine law models. These standard creep laws are used for modeling secondary or steady-state creep. Creep is defined by including creep behavior in the material model definition ("Material data definition," Section 21.1.2). Alternatively, creep can be defined in conjunction with gasket behavior to define the rate-dependent behavior of a gasket.

Input File Usage: Use the following options to include creep behavior in the material model definition:

```
*MATERIAL
*CREEP
```

Use the following options to define creep in conjunction with gasket behavior:

```
*GASKET BEHAVIOR
*CREEP
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Creep**

Choosing a creep model

The power-law creep model is attractive for its simplicity. However, it is limited in its range of application. The time-hardening version of the power-law creep model is typically recommended only in cases when the stress state remains essentially constant. The strain-hardening version of power-law creep should be used when the stress state varies during an analysis. In the case where the stress is constant and there are no temperature and/or field dependencies, the time-hardening and

strain-hardening versions of the power-creep law are equivalent. For either version of the power law, the stresses should be relatively low.

In regions of high stress, such as around a crack tip, the creep strain rates frequently show an exponential dependence of stress. The hyperbolic-sine creep law shows exponential dependence on the stress, σ, at high stress levels ($\sigma/\sigma^0 \gg 1$, where σ^0 is the yield stress) and reduces to the power-law at low stress levels (with no explicit time dependence).

None of the above models is suitable for modeling creep under cyclic loading. The ORNL model ("ORNL – Oak Ridge National Laboratory constitutive model," Section 23.2.12) is an empirical model for stainless steel that gives approximate results for cyclic loading without having to perform the cyclic loading numerically. Generally, creep models for cyclic loading are complicated and must be added to a model with user subroutine **CREEP** or with user subroutine **UMAT**.

Modeling simultaneous creep and plasticity

If creep and plasticity occur simultaneously and implicit creep integration is in effect, both behaviors may interact and a coupled system of constitutive equations needs to be solved. If creep and plasticity are isotropic, Abaqus/Standard properly takes into account such coupled behavior, even if the elasticity is anisotropic. However, if creep and plasticity are anisotropic, Abaqus/Standard integrates the creep equations without taking plasticity into account, which may lead to substantial errors in the creep strains. This situation develops only if plasticity and creep are active at the same time, such as would occur during a long-term load increase; one would not expect to have a problem if there is a short-term preloading phase in which plasticity dominates, followed by a creeping phase in which no further yielding occurs. Integration of the creep laws and rate-dependent plasticity are discussed in "Rate-dependent metal plasticity (creep)," Section 4.3.4 of the Abaqus Theory Manual.

Power-law model

The power-law model can be used in its "time hardening" form or in the corresponding "strain hardening" form.

Time hardening form

The "time hardening" form is the simpler of the two forms of the power-law model:

$$\dot{\bar{\varepsilon}}^{cr} = A\tilde{q}^n t^m,$$

where

$\dot{\bar{\varepsilon}}^{cr}$	is the uniaxial equivalent creep strain rate, $\sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}$,
$\tilde{q}$	is the uniaxial equivalent deviatoric stress,
t	is the total time, and
A, n, and m	are defined by you as functions of temperature.

$\tilde{q}$ is Mises equivalent stress or Hill's anisotropic equivalent deviatoric stress according to whether isotropic or anisotropic creep behavior is defined (discussed below). For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$. Since total time is used in the expression, such reasonable

behavior also typically requires that small step times compared to the creep time be used for any steps for which creep is not active in an analysis; this is necessary to avoid changes in hardening behavior in subsequent steps.

Input File Usage: *CREEP, LAW=TIME

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Creep**: **Law: Time-Hardening**

Strain hardening form

The "strain hardening" form of the power law is

$$\dot{\bar{\varepsilon}}^{cr} = \left(A\tilde{q}^n [(m+1)\bar{\varepsilon}^{cr}]^m \right)^{\frac{1}{m+1}},$$

where $\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$ are defined above and $\bar{\varepsilon}^{cr}$ is the equivalent creep strain.

Input File Usage: *CREEP, LAW=STRAIN

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Creep**: **Law: Strain-Hardening**

Numerical difficulties

Depending on the choice of units for either form of the power law, the value of A may be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations; therefore, use another system of units to avoid such difficulties in the calculation of creep strain increments.

Hyperbolic-sine law model

The hyperbolic-sine law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A(\sinh B\tilde{q})^n \exp\left(-\frac{\triangle H}{R(\theta - \theta^Z)} \right),$$

where

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$	are defined above,
θ	is the temperature,
θ^Z	is the user-defined value of absolute zero on the temperature scale used,
$\triangle H$	is the activation energy,
R	is the universal gas constant, and
A, B, and n	are other material parameters.

This model includes temperature dependence, which is apparent in the above expression; however, the parameters A, B, n, $\triangle H$, and R cannot be defined as functions of temperature.

Input File Usage: Use both of the following options:

*CREEP, LAW=HYPERB
*PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Define both of the following:

Property module: material editor: **Mechanical→Plasticity→Creep**: **Law: Hyperbolic-Sine**

Any module: **Model→Edit Attributes→*model_name***: **Absolute zero temperature**

Numerical difficulties

As with the power law, A may be very small for typical creep strain rates. If A is very small (such as less than 10^{-27}), use another system of units to avoid numerical difficulties in the calculation of creep strain increments.

Anisotropic creep

Anisotropic creep can be defined to specify the stress ratios that appear in Hill's function. You must define the ratios R_{ij} in each direction that will be used to scale the stress value when the creep strain rate is calculated. The ratios can be defined as constant or dependent on temperature and other predefined field variables. The ratios are defined with respect to the user-defined local material directions or the default directions (see "Orientations," Section 2.2.5). Further details are provided in "Anisotropic yield/creep," Section 23.2.6. Anisotropic creep is not available when creep is used to define a rate-dependent gasket behavior since only the gasket thickness-direction behavior can have rate-dependent behavior.

Input File Usage: *POTENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Creep**: **Suboptions→Potential**

Volumetric swelling behavior

As with the creep laws, volumetric swelling laws are usually complex and are most conveniently specified in user subroutine **CREEP** as discussed below. However, a means of tabular input is also provided for the form

$$\dot{\bar{\varepsilon}}^{sw} = f(\theta, f_1, f_2, \ldots),$$

where $\dot{\bar{\varepsilon}}^{sw}$ is the volumetric strain rate caused by swelling and f_1, f_2, ... are predefined fields such as irradiation fluxes in cases involving nuclear radiation effects. Up to six predefined fields can be specified.

Volumetric swelling cannot be used to define a rate-dependent gasket behavior.

Input File Usage: *SWELLING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Swelling**

Anisotropic swelling

Anisotropy can easily be included in the swelling behavior. If anisotropic swelling behavior is defined, the anisotropic swelling strain rate is expressed as

$$\dot{\bar{\varepsilon}}_A^{sw} = \dot{\bar{\varepsilon}}_{11}^{sw} + \dot{\bar{\varepsilon}}_{22}^{sw} + \dot{\bar{\varepsilon}}_{33}^{sw} = (r_{11} + r_{22} + r_{33})\frac{1}{3}\dot{\bar{\varepsilon}}^{sw},$$

where $\dot{\bar{\varepsilon}}^{sw}$ is the volumetric swelling strain rate that you define either directly (discussed above) or in user subroutine **CREEP**. The ratios r_{11}, r_{22}, and r_{33} are also user-defined. The directions of the components of the swelling strain rate are defined by the local material directions, which can be either user-defined or the default directions (see "Orientations," Section 2.2.5).

Input File Usage: Use both of the following options:

*SWELLING
*RATIOS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Swelling**: **Suboptions→Ratios**

User subroutine CREEP

User subroutine **CREEP** provides a very general capability for implementing viscoplastic models such as creep and swelling models in which the strain rate potential can be written as a function of equivalent pressure stress, p; the Mises or Hill's equivalent deviatoric stress, $\tilde{q}$; and any number of solution-dependent state variables. Solution-dependent state variables are used in conjunction with the constitutive definition; their values evolve with the solution and can be defined in this subroutine. Examples are hardening variables associated with the model.

The user subroutine can also be used to define very general rate- and time-dependent thickness-direction gasket behavior. When an even more general form is required for the strain rate potential, user subroutine **UMAT** ("User-defined mechanical material behavior," Section 26.7.1) can be used.

Input File Usage: Use one or both of the following options. Only the first option can be used to define gasket behavior.

*CREEP, LAW=USER
*SWELLING, LAW=USER

Abaqus/CAE Usage: Use one or both of the following models. Only the first model can be used to define gasket behavior.

Property module: material editor:
Mechanical→Plasticity→Creep: **Law: User defined**
Mechanical→Plasticity→Swelling: **Law: User subroutine CREEP**

Removing creep effects in an analysis step

You can specify that no creep (or viscoelastic) response can occur during certain analysis steps, even if creep (or viscoelastic) material properties have been defined.

Input File Usage: Use one of the following options:

*COUPLED TEMPERATURE-DISPLACEMENT, CREEP=NONE
*SOILS, CONSOLIDATION, CREEP=NONE

Abaqus/CAE Usage: Use one of the following options:

Step module: **Create Step**:
Coupled temp-displacement: toggle off **Include creep/swelling/viscoelastic behavior**

Soils: **Pore fluid response: Transient consolidation**: toggle off **Include creep/swelling/viscoelastic behavior**

Integration

Explicit integration, implicit integration, or both integration schemes can be used in a creep analysis, depending on the procedure used, the parameters specified for the procedure, the presence of plasticity, and whether or not geometric nonlinearity is requested.

Application of explicit and implicit schemes

Nonlinear creep problems are often solved efficiently by forward-difference integration of the inelastic strains (the "initial strain" method). This explicit method is computationally efficient because, unlike implicit methods, iteration is not required. Although this method is only conditionally stable, the numerical stability limit of the explicit operator is usually sufficiently large to allow the solution to be developed in a small number of time increments.

Abaqus/Standard uses either an explicit or an implicit integration scheme or switches from explicit to implicit in the same step. These schemes are outlined first, followed by a description of which procedures use these integration schemes.

1. Integration scheme 1: Starts with explicit integration and switches to implicit integration based on either stability or if plasticity is active. The stability limit used in explicit integration is discussed in the next section.
2. Integration scheme 2: Starts with explicit integration and switches to implicit integration when plasticity is active. The stability criterion does not play a role here.
3. Integration scheme 3: Always uses implicit integration.

The use of the above integration schemes is determined by the procedure type, your choice of the integration type to be used, as well as whether or not geometric nonlinearity is requested. For quasi-static and coupled temperature-displacement procedures, if you do not choose an integration type, integration scheme 1 is used for a geometrically linear analysis and integration scheme 3 is used for a geometrically nonlinear analysis. You can force Abaqus/Standard to use explicit integration for creep and

swelling effects in coupled temperature-displacement or quasi-static procedures, when plasticity is not active throughout the step (integration scheme 2). Explicit integration can be used regardless of whether or not geometric nonlinearity has been requested (see "General and linear perturbation procedures," Section 6.1.3).

For a transient soils consolidation procedure, the implicit integration scheme (integration scheme 3) is always used, irrespective of whether a geometrically linear or nonlinear analysis is performed.

Input File Usage: Use one of the following options to restrict Abaqus/Standard to using explicit integration:

*VISCO, CREEP=EXPLICIT
*COUPLED TEMPERATURE-DISPLACEMENT, CREEP=EXPLICIT

Abaqus/CAE Usage: Use one of the following options to restrict Abaqus/Standard to using explicit integration:

Step module: **Create Step**:
Visco: **Incrementation**: **Creep/swelling/viscoelastic integration: Explicit**

Coupled temp-displacement: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: **Creep/swelling/viscoelastic integration: Explicit**

Automatic monitoring of stability limit during explicit integration

Abaqus/Standard monitors the stability limit automatically during explicit integration. If, at any point in the model, the creep strain increment ($\dot{\bar{\varepsilon}}^{cr}|_t \Delta t$) is larger than the total elastic strain, the problem will become unstable. Therefore, a stable time step, Δt_s, is calculated every increment by

$$\Delta t_s = 0.5 \frac{\varepsilon^{el}|_t}{\dot{\bar{\varepsilon}}^{cr}|_t},$$

where $\varepsilon^{el}|_t$ is the equivalent total elastic strain at time t, the beginning of the increment, and $\dot{\bar{\varepsilon}}^{cr}|_t$ is the equivalent creep strain rate at time t. Furthermore,

$$\varepsilon^{el}|_t = \frac{\tilde{q}|_t}{\tilde{E}},$$

where $\tilde{q}|_t$ is the Mises stress at time t, and

$$\tilde{E} = 2(1+\nu)(\mathbf{n} : \mathbf{D}^{el} : \mathbf{n}) \approx 2.5\bar{E},$$

where

$\mathbf{n} = \partial \tilde{q}|_t / \partial \boldsymbol{\sigma}$ is the gradient of the deviatoric stress potential,

$\mathbf{D}^{el}$ is the elasticity matrix, and

$\bar{E}$ is an effective elastic modulus—for isotropic elasticity $\bar{E}$ can be approximated by Young's modulus.

At every increment for which explicit integration is performed, the stable time increment, Δt_s, is compared to the critical time increment, Δt_c, which is calculated as follows:

$$\Delta t_c = \frac{errtol}{\dot{\bar{\varepsilon}}^{cr}|_{t+\Delta t} - \dot{\bar{\varepsilon}}^{cr}|_t}.$$

The quantity *errtol* is an error tolerance that you define as discussed below. If Δt_s is less than Δt_c, Δt_s is used as the time increment, which would mean that the stability criterion was limiting the size of the time step further than required by accuracy considerations. Abaqus/Standard will automatically switch to the backward difference operator (the implicit method, which is unconditionally stable) if Δt_s is less than Δt_c for nine consecutive increments, you have not restricted Abaqus/Standard to explicit integration as discussed above, and there is sufficient time left in the analysis (time left $\geq 50\Delta t$). The stiffness matrix will be reformed at every iteration if the implicit algorithm is used.

Specifying the tolerance for automatic incrementation

The integration tolerance must be chosen so that increments in stress, $\triangle\boldsymbol{\sigma}$, are calculated accurately. Consider a one-dimensional example. The stress increment, $\triangle\sigma$, is

$$\triangle\sigma = E\triangle\varepsilon^{el} = E(\triangle\varepsilon - \triangle\varepsilon^{cr}),$$

where $\triangle\varepsilon^{el}$, $\triangle\varepsilon$, and $\triangle\varepsilon^{cr}$ are the uniaxial elastic, total, and creep strain increments, respectively, and $\boldsymbol{E}$ is the elastic modulus. For $\triangle\sigma$ to be calculated accurately, the error in the creep strain increment, $\triangle\varepsilon^{cr}_{err}$, must be small compared to $\triangle\varepsilon^{el}$; that is,

$$\triangle\varepsilon^{cr}_{err} \ll \triangle\varepsilon^{el}.$$

Measuring the error in $\triangle\varepsilon^{cr}$ as

$$\triangle\varepsilon^{cr}_{err} = (\dot{\bar{\varepsilon}}^{cr}|_{t+\triangle t} - \dot{\bar{\varepsilon}}^{cr}|_t)\triangle t$$

leads to

$$(\dot{\bar{\varepsilon}}^{cr}|_{t+\triangle t} - \dot{\bar{\varepsilon}}^{cr}|_t)\triangle t \ll \triangle\varepsilon^{el} = \frac{\triangle\sigma}{E}, \text{ or}$$

$$errtol \ll \frac{\triangle\sigma}{E}.$$

You define *errtol* for the applicable procedure by choosing an acceptable stress error tolerance and dividing this by a typical elastic modulus; therefore, it should be a small fraction of the ratio of the typical stress and the effective elastic modulus in a problem. It is important to recognize that this approach for selecting a value for *errtol* is often very conservative, and acceptable solutions can usually be obtained with higher values.

Input File Usage: Use one of the following options:

*VISCO, CETOL=*errtol*
*COUPLED TEMPERATURE-DISPLACEMENT, CETOL=*errtol*
*SOILS, CONSOLIDATION, CETOL=*errtol*

Abaqus/CAE Usage: Use one of the following options:

Step module: **Create Step**:
Visco: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Coupled temp-displacement: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Soils: **Pore fluid response: Transient consolidation**: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Loading control using creep strain rate

In superplastic forming a controllable pressure is applied to deform a body. Superplastic materials can deform to very large strains, provided that the strain rates of the deformation are maintained within very tight tolerances. The objective of the superplastic analysis is to predict how the pressure must be controlled to form the component as fast as possible without exceeding a superplastic strain rate anywhere in the material.

To achieve this using Abaqus/Standard, the controlling algorithm is as follows. During an increment Abaqus/Standard calculates $r_{\max}$, the maximum value of the ratio of the equivalent creep strain rate to the target creep strain rate for any integration point in a specified element set. If $r_{\max}$ is less than 0.2 or greater than 3.0 in a given increment, the increment is abandoned and restarted with the following load modifications:

$$\begin{aligned} r_{\max} &< 0.2 \qquad p = 2.0\, p_{\text{old}}, \text{ or} \\ r_{\max} &> 3.0 \qquad p = 0.5\, p_{\text{old}}, \end{aligned}$$

where p is the new load magnitude and p_{old} is the old load magnitude. If $0.2 \le r_{\max} \le 3.0$, the increment is accepted; and at the beginning of the following time increment, the load magnitudes are modified as follows:

$$\begin{aligned} 0.2 \le r_{\max} &< 0.5 \qquad p = 1.5\, p_{\text{old}}; \\ 0.5 \le r_{\max} &< 0.8 \qquad p = 1.2\, p_{\text{old}}; \\ 0.8 \le r_{\max} &< 1.5 \qquad p = p_{\text{old}}; \text{ or} \\ 1.5 \le r_{\max} &\le 3.0 \qquad p = p_{\text{old}}/1.2. \end{aligned}$$

When you activate the above algorithm, the loading in a creep and/or swelling problem can be controlled on the basis of the maximum equivalent creep strain rate found in a defined element set. As

a minimum requirement, this method is used to define a target equivalent creep strain rate; however, if required, it can also be used to define the target creep strain rate as a function of equivalent creep strain (measured as log strain), temperature, and other predefined field variables. The creep strain dependency curve at each temperature must always start at zero equivalent creep strain.

A solution-dependent amplitude is used to define the minimum and maximum limits of the loading (see "Defining a solution-dependent amplitude for superplastic forming analysis" in "Amplitude curves," Section 33.1.2). Any number or combination of loads can be used. The current value of $r_{\max}$ is available for output as discussed below.

Input File Usage: Use all of the following options:

∗AMPLITUDE, NAME=*name*, DEFINITION=SOLUTION DEPENDENT
∗CLOAD, ∗DLOAD, ∗DSLOAD, and/or ∗BOUNDARY with AMPLITUDE=*name*
∗CREEP STRAIN RATE CONTROL, AMPLITUDE=*name*, ELSET=*elset*

The ∗AMPLITUDE option must appear in the model definition portion of an input file, while the loading options (∗CLOAD, ∗DLOAD, ∗DSLOAD, and ∗BOUNDARY) and the ∗CREEP STRAIN RATE CONTROL option should appear in each relevant step definition.

Abaqus/CAE Usage: Creep strain rate control is not supported in Abaqus/CAE.

Elements

Rate-dependent plasticity (creep and swelling behavior) can be used with any continuum, shell, membrane, gasket, and beam element in Abaqus/Standard that has displacement degrees of freedom. Creep (but not swelling) can also be defined in the thickness direction of any gasket element in conjunction with the gasket behavior definition.

Output

In addition to the standard output identifiers available in Abaqus/Standard ("Abaqus/Standard output variable identifiers," Section 4.2.1), the following variables relate directly to creep and swelling models:

CEEQ	Equivalent creep strain, $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}\, dt$.
CESW	Magnitude of swelling strain.

The following output, which is relevant only for an analysis with creep strain rate loading control as discussed above, is printed at the beginning of an increment and is written automatically to the results file and output database file when any output to these files is requested:

RATIO	Maximum value of the ratio of the equivalent creep strain rate to the target creep strain rate, r_{max}.
AMPCU	Current value of the solution-dependent amplitude.

23.2.5 ANNEALING OR MELTING

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- *ANNEAL TEMPERATURE
- "Specifying the annealing temperature of an elastic-plastic material" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

This capability:

- is intended to model the effects of melting and resolidification in metals subjected to high-temperature processes or the effects of annealing at a material point when its temperature rises above a certain level;
- is available for only the Mises, Johnson-Cook, and Hill plasticity models;
- is intended to be used in conjunction with appropriate temperature-dependent material properties (in particular, the model assumes perfectly plastic behavior at or above the annealing or melting temperature); and
- can be modeled simply by defining an annealing or melting temperature.

Effects of annealing or melting

When the temperature of a material point exceeds a user-specified value called the annealing temperature, Abaqus assumes that the material point loses its hardening memory. The effect of prior work hardening is removed by setting the equivalent plastic strain to zero. For kinematic and combined hardening models the backstress tensor is also reset to zero. If the temperature of the material point falls below the annealing temperature at a subsequent point in time, the material point can work harden again. Depending on the temperature history a material point may lose and accumulate memory several times, which in the context of modeling melting would correspond to repeated melting and resolidification. Any accumulated material damage is not healed when the annealing temperature is reached. Damage will continue to accumulate after annealing according to any damage model in effect (see "Damage and failure for ductile metals: overview," Section 24.2.1).

In Abaqus/Explicit an annealing step can be defined to simulate the annealing process for the entire model, independent of temperature; see "Annealing procedure," Section 6.12.1, for details.

Material properties

The annealing temperature is a material property that can optionally be defined as a function of field variables. This material property must be used in conjunction with an appropriate definition of material

properties as functions of temperature for the Mises plasticity model. In particular, the hardening behavior must be defined as a function of temperature and zero hardening must be specified at or above the annealing temperature. In general, hardening receives contributions from two sources. The first source of hardening can be classified broadly as static, and its effect is measured by the rate of change of the yield stress with respect to the plastic strain at a fixed strain rate. The second source of hardening can be classified broadly as rate dependent, and its effect is measured by the rate of change of the yield stress with respect to the strain rate at a fixed plastic strain.

For the Mises plasticity model, if the material data that describe hardening (both static and rate-dependent contributions) are completely specified through tabular input of yield stress versus plastic strain at different values of the strain rate (see "Rate-dependent yield," Section 23.2.3), the (temperature-dependent) static part of the hardening at each strain rate is specified by defining several yield stress versus plastic strain curves (each at a different temperature). For metals the yield stress at a fixed strain rate typically decreases with increasing temperature. Abaqus expects the hardening at each strain rate to vanish at or above the annealing temperature and issues an error message if you specify otherwise in the material definition. Zero (static) hardening can be specified by simply specifying a single data point (at zero plastic strain) in the yield stress versus plastic strain curve at or above the annealing temperature. In addition, you must also ensure that at or above the annealing temperature, the yield stress does not vary with the strain rate. This can be accomplished by specifying the same value of yield stress at all values of strain rate in the single data point approach discussed above.

Alternatively, the static part of the hardening can be defined at zero strain rate, and the rate-dependent part can be defined utilizing the overstress power law (see "Rate-dependent yield," Section 23.2.3). In that case, zero static hardening at or above the annealing temperature can be specified by specifying a single data point (at zero plastic strain) in the yield stress versus plastic strain curve at or above the annealing temperature. The overstress power law parameters can also be appropriately selected to ensure that at or above the annealing temperature the yield stress does not vary with strain rate. This can be accomplished by selecting a large value for the parameter D (relative to the static yield stress) and setting the parameter $n = 1$.

For hardening defined in Abaqus/Standard with user subroutine **UHARD**, Abaqus/Standard checks the hardening slope at or above the annealing temperature during the actual computations and issues an error message if appropriate.

The Johnson-Cook plasticity model in Abaqus/Explicit requires a separate melting temperature to define the hardening behavior. If the annealing temperature is defined to be less than the melting temperature specified for the metal plasticity model, the hardening memory is removed at the annealing temperature and the melting temperature is used strictly to define the hardening function. Otherwise, the hardening memory is removed automatically at the melting temperature.

Input File Usage: *ANNEAL TEMPERATURE

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Anneal Temperature**

Example: Annealing or melting

The following input is an example of a typical usage of the annealing or melting capability. It is assumed that you have defined the static stress versus plastic strain behavior (see Figure 23.2.5–1) for the isotropic

hardening model at three different temperatures, including the annealing temperature. It is also assumed that the plastic behavior is rate independent.

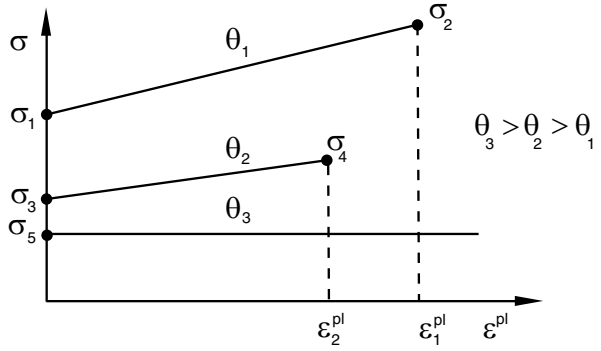


Figure 23.2.5–1 Stress versus plastic strain behavior.

The plastic response corresponds to linear hardening below the annealing temperature and perfect plasticity at the annealing temperature. The elastic properties, which may also be temperature dependent, are not shown.

Plasticity Data, Isotropic Hardening:		
Yield Stress	**Plastic Strain**	**Temperature**
σ_1	0	θ_1
σ_2	ϵ_1^{pl}	θ_1
σ_3	0	θ_2
σ_4	ϵ_2^{pl}	θ_2
σ_5	0	θ_3
Anneal Temperature: θ_3		

Elements

This capability can be used with all elements that include mechanical behavior (elements that have displacement degrees of freedom).

Output

Only the equivalent plastic strain (output variable PEEQ) and the backstress (output variable ALPHA) are reset to zero at the melting temperature. The plastic strain tensor (output variable PE) is not reset to

zero and provides a measure of the total plastic deformation during the analysis. In Abaqus/Standard the plastic strain tensor also provides a measure of the plastic strain magnitude (output variable PEMAG).

23.2.6 ANISOTROPIC YIELD/CREEP

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "Classical metal plasticity," Section 23.2.1
- "Models for metals subjected to cyclic loading," Section 23.2.2
- "Rate-dependent plasticity: creep and swelling," Section 23.2.4
- *POTENTIAL
- "Defining anisotropic yield and creep" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Anisotropic yield and/or creep:

- can be used for materials that exhibit different yield and/or creep behavior in different directions;
- is introduced through user-defined stress ratios that are applied in Hill's potential function;
- can be used only in conjunction with the metal plasticity and, in Abaqus/Standard, the metal creep material models;
- is available for the nonlinear isotropic/kinematic hardening model in Abaqus/Explicit ("Models for metals subjected to cyclic loading," Section 23.2.2); and
- can be used in conjunction with the models of progressive damage and failure in Abaqus/Explicit ("Damage and failure for ductile metals: overview," Section 24.2.1) to specify different damage initiation criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh.

Yield and creep stress ratios

Anisotropic yield or creep behavior is modeled through the use of yield or creep stress ratios, R_{ij}. In the case of anisotropic yield the yield ratios are defined with respect to a reference yield stress, σ^0 (given for the metal plasticity definition), such that if σ_{ij} is applied as the only nonzero stress, the corresponding yield stress is $R_{ij}\sigma^0$. The plastic flow rule is defined below.

In the case of anisotropic creep the R_{ij} are creep ratios used to scale the stress value when the creep strain rate is calculated. Thus, if σ_{11} is the only nonzero stress, the equivalent stress, $\tilde{q}$, used in the user-defined creep law is $\tilde{q} = R_{11}|\sigma_{11}|$.

Yield and creep stress ratios can be defined as constants or as tabular functions of temperature and predefined field variables. A local orientation must be used to define the direction of anisotropy (see "Orientations," Section 2.2.5).

Input File Usage: Use the following option to define the yield or creep stress ratios:

*POTENTIAL

This option must appear immediately after the *PLASTIC or the *CREEP material option data to which it applies. Thus, if anisotropic metal plasticity and anisotropic creep behavior are both required, the *POTENTIAL option must appear twice in the material definition, once after the metal plasticity data and again after the creep data.

Abaqus/CAE Usage: Use one of the following models:

Property module: material editor:
Mechanical→Plasticity→Plastic: **Suboptions→Potential**
Mechanical→Plasticity→Creep: **Suboptions→Potential**

Anisotropic yield

Hill's potential function is a simple extension of the Mises function, which can be expressed in terms of rectangular Cartesian stress components as

$$f(\boldsymbol{\sigma}) = \sqrt{F(\sigma_{22}-\sigma_{33})^2 + G(\sigma_{33}-\sigma_{11})^2 + H(\sigma_{11}-\sigma_{22})^2 + 2L{\sigma_{23}}^2 + 2M{\sigma_{31}}^2 + 2N{\sigma_{12}}^2},$$

where $F, G, H, L, M,$ and N are constants obtained by tests of the material in different orientations. They are defined as

$$F = \frac{(\sigma^0)^2}{2}\left(\frac{1}{\bar{\sigma}_{22}^2} + \frac{1}{\bar{\sigma}_{33}^2} - \frac{1}{\bar{\sigma}_{11}^2}\right) = \frac{1}{2}\left(\frac{1}{R_{22}^2} + \frac{1}{R_{33}^2} - \frac{1}{R_{11}^2}\right),$$

$$G = \frac{(\sigma^0)^2}{2}\left(\frac{1}{\bar{\sigma}_{33}^2} + \frac{1}{\bar{\sigma}_{11}^2} - \frac{1}{\bar{\sigma}_{22}^2}\right) = \frac{1}{2}\left(\frac{1}{R_{33}^2} + \frac{1}{R_{11}^2} - \frac{1}{R_{22}^2}\right),$$

$$H = \frac{(\sigma^0)^2}{2}\left(\frac{1}{\bar{\sigma}_{11}^2} + \frac{1}{\bar{\sigma}_{22}^2} - \frac{1}{\bar{\sigma}_{33}^2}\right) = \frac{1}{2}\left(\frac{1}{R_{11}^2} + \frac{1}{R_{22}^2} - \frac{1}{R_{33}^2}\right),$$

$$L = \frac{3}{2}\left(\frac{\tau^0}{\bar{\sigma}_{23}}\right)^2 = \frac{3}{2R_{23}^2},$$

$$M = \frac{3}{2}\left(\frac{\tau^0}{\bar{\sigma}_{13}}\right)^2 = \frac{3}{2R_{13}^2},$$

$$N = \frac{3}{2}\left(\frac{\tau^0}{\bar{\sigma}_{12}}\right)^2 = \frac{3}{2R_{12}^2},$$

where each $\bar{\sigma}_{ij}$ is the measured yield stress value when σ_{ij} is applied as the only nonzero stress component; σ^0 is the user-defined reference yield stress specified for the metal plasticity definition; R_{11}, R_{22}, R_{33}, R_{12}, R_{13}, and R_{23} are anisotropic yield stress ratios; and $\tau^0 = \sigma^0/\sqrt{3}$. The six yield stress ratios are, therefore, defined as follows (in the order in which you must provide them):

$$\frac{\bar{\sigma}_{11}}{\sigma^0}, \quad \frac{\bar{\sigma}_{22}}{\sigma^0}, \quad \frac{\bar{\sigma}_{33}}{\sigma^0}, \quad \frac{\bar{\sigma}_{12}}{\tau^0}, \quad \frac{\bar{\sigma}_{13}}{\tau^0}, \quad \frac{\bar{\sigma}_{23}}{\tau^0}.$$

Because of the form of the yield function, all of these ratios must be positive. If the constants F, G, and H are positive, the yield function is always well-defined. However, if one or more of these constants is negative, the yield function may be undefined for some stress states because the quantity under the square root is negative.

The flow rule is

$$d\boldsymbol{\varepsilon}^{pl} = d\lambda \frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{d\lambda}{f}\mathbf{b},$$

where, from the definition of f above,

$$\mathbf{b} = \begin{bmatrix} -G(\sigma_{33} - \sigma_{11}) + H(\sigma_{11} - \sigma_{22}) \\ F(\sigma_{22} - \sigma_{33}) - H(\sigma_{11} - \sigma_{22}) \\ -F(\sigma_{22} - \sigma_{33}) + G(\sigma_{33} - \sigma_{11}) \\ 2N\sigma_{12} \\ 2M\sigma_{31} \\ 2L\sigma_{23} \end{bmatrix}.$$

Input File Usage: Use both of the following options:
*PLASTIC
*POTENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Potential**

Anisotropic creep

For anisotropic creep in Abaqus/Standard Hill's function can be expressed as

$$\tilde{q}(\boldsymbol{\sigma}) = \sqrt{F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L{\sigma_{23}}^2 + 2M{\sigma_{31}}^2 + 2N{\sigma_{12}}^2},$$

where $\tilde{q}(\boldsymbol{\sigma})$ is the equivalent stress and F, G, H, L, M, and N are constants obtained by tests of the material in different orientations. The constants are defined with the same general relations as those used for anisotropic yield (above); however, the reference yield stress, σ^0, is replaced by the uniaxial equivalent deviatoric stress, $\tilde{q}$ (found in the creep law), and R_{11}, R_{22}, R_{33}, R_{12}, R_{13}, and R_{23} are referred

to as "anisotropic creep stress ratios." The six creep stress ratios are, therefore, defined as follows (in the order in which they must be provided):

$$\frac{\sigma_{11}}{\tilde{q}}, \quad \frac{\sigma_{22}}{\tilde{q}}, \quad \frac{\sigma_{33}}{\tilde{q}}, \quad \frac{\sigma_{12}}{\tilde{q}/\sqrt{3}}, \quad \frac{\sigma_{13}}{\tilde{q}/\sqrt{3}}, \quad \frac{\sigma_{23}}{\tilde{q}/\sqrt{3}}.$$

You must define the ratios R_{ij} in each direction that will be used to scale the stress value when the creep strain rate is calculated. If all six R_{ij} values are set to unity, isotropic creep is obtained.

Input File Usage: Use both of the following options:

∗CREEP
∗POTENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Creep**: **Suboptions→Potential**

Defining anisotropic yield behavior on the basis of strain ratios (Lankford's r-values)

As discussed above, Hill's anisotropic plasticity potential is defined in Abaqus from user input consisting of ratios of yield stress in different directions with respect to a reference stress. However, in some cases, such as sheet metal forming applications, it is common to find the anisotropic material data given in terms of ratios of width strain to thickness strain. Mathematical relationships are then necessary to convert the strain ratios to stress ratios that can be input into Abaqus.

In sheet metal forming applications we are generally concerned with plane stress conditions. Consider x, y to be the "rolling" and "cross" directions in the plane of the sheet; z is the thickness direction. From a design viewpoint, the type of anisotropy usually desired is that in which the sheet is isotropic in the plane and has an increased strength in the thickness direction, which is normally referred to as transverse anisotropy. Another type of anisotropy is characterized by different strengths in different directions in the plane of the sheet, which is called planar anisotropy.

In a simple tension test performed in the x-direction in the plane of the sheet, the flow rule for this potential (given above) defines the incremental strain ratios (assuming small elastic strains) as

$$d\varepsilon_{11} \ : \ d\varepsilon_{22} \ : \ d\varepsilon_{33} = G + H \ : \ -H \ : \ -G.$$

Therefore, the ratio of width to thickness strain, often referred to as Lankford's r-value, is

$$r_x = \frac{d\varepsilon_{22}}{d\varepsilon_{33}} = \frac{H}{G}.$$

Similarly, for a simple tension test performed in the y-direction in the plane of the sheet, the incremental strain ratios are

$$d\varepsilon_{11} \ : \ d\varepsilon_{22} \ : \ d\varepsilon_{33} = -H \ : \ F + H \ : \ -F,$$

and

$$r_y = \frac{d\varepsilon_{11}}{d\varepsilon_{33}} = \frac{H}{F}.$$

Transverse anisotropy

A transversely anisotropic material is one where $r_x = r_y$. If we define σ^0 in the metal plasticity model to be equal to $\bar{\sigma}_{11}$,

$$R_{11} = R_{22} = 1$$

and, using the relationships above,

$$R_{33} = \sqrt{\frac{r_x + 1}{2}}.$$

If $r_x = 1$ (isotropic material), $R_{33} = 1$ and the Mises isotropic plasticity model is recovered.

Planar anisotropy

In the case of planar anisotropy r_x and r_y are different and R_{11}, R_{22}, R_{33} will all be different. If we define σ^0 in the metal plasticity model to be equal to $\bar{\sigma}_{11}$,

$$R_{11} = 1$$

and, using the relationships above, we obtain

$$R_{22} = \sqrt{\frac{r_y(r_x + 1)}{r_x(r_y + 1)}}, \qquad R_{33} = \sqrt{\frac{r_y(r_x + 1)}{(r_x + r_y)}}.$$

Again, if $r_x = r_y = 1$, $R_{22} = R_{33} = 1$ and the Mises isotropic plasticity model is recovered.

General anisotropy

Thus far, we have only considered loading applied along the axes of anisotropy. To derive a more general anisotropic model in plane stress, the sheet must be loaded in one other direction in its plane. Suppose we perform a simple tension test at an angle α to the x-direction; then, from equilibrium considerations we can write the nonzero stress components as

$$\sigma_{11} = \sigma \cos^2\alpha, \quad \sigma_{22} = \sigma \sin^2\alpha, \quad \sigma_{12} = \sigma \sin\alpha \cos\alpha,$$

where σ is the applied tensile stress. Substituting these values in the flow equations and assuming small elastic strains yields

$$d\varepsilon_{11} = [(G + H)\cos^2\alpha - H \sin^2\alpha]\frac{\sigma}{f}\, d\lambda,$$

$$d\varepsilon_{22} = [(F+H)\sin^2\alpha - H\cos^2\alpha]\frac{\sigma}{f}\,d\lambda,$$

$$d\varepsilon_{33} = -[F\sin^2\alpha + G\cos^2\alpha]\frac{\sigma}{f}\,d\lambda, \text{ and}$$

$$d\gamma_{12} = [N\sin\alpha\cos\alpha]\frac{\sigma}{f}\,d\lambda.$$

Assuming small geometrical changes, the width strain increment (the increment of strain at right angles to the direction of loading, α) is written as

$$d\varepsilon_{\alpha+\frac{\pi}{2}} = d\varepsilon_{11}\sin^2\alpha + d\varepsilon_{22}\cos^2\alpha - 2\,d\gamma_{12}\sin\alpha\cos\alpha,$$

and Lankford's r-value for loading at an angle α is

$$r_\alpha = \frac{d\varepsilon_{\alpha+\frac{\pi}{2}}}{d\varepsilon_{33}} = \frac{H + (2N - F - G - 4H)\sin^2\alpha\cos^2\alpha}{F\sin^2\alpha + G\cos^2\alpha}.$$

One of the more commonly performed tests is that in which the loading direction is at 45°. In this case

$$r_{45} = \frac{2N - (F+G)}{2(F+G)} \quad \text{or} \quad \frac{N}{G} = (r_{45} + \frac{1}{2})(1 + \frac{r_x}{r_y}).$$

If σ^0 is equal to $\bar{\sigma}_{11}$ in the metal plasticity model, $R_{11} = 1$. R_{22}, R_{33} are as defined before for transverse or planar anisotropy and, using the relationships above,

$$R_{12} = \sqrt{\frac{3(r_x+1)r_y}{(2r_{45}+1)(r_x+r_y)}}.$$

Progressive damage and failure

In Abaqus/Explicit anisotropic yield can be used in conjunction with the models of progressive damage and failure discussed in "Damage and failure for ductile metals: overview," Section 24.2.1. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), and Müschenborn-Sonne forming limit diagram (MSFLD) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations.

Input File Usage: Use the following options:

*PLASTIC

*DAMAGE INITIATION
*DAMAGE EVOLUTION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*damage initiation type***: specify the damage initiation criterion: **Suboptions→Damage Evolution**: specify the damage evolution parameters

Initial conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, Abaqus allows you to prescribe initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, by specifying the conditions directly ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

User subroutine specification in Abaqus/Standard

For more complicated cases, initial conditions can be defined in Abaqus/Standard through user subroutine **HARDINI**.

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING, USER

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

Anisotropic yield can be defined for any element that can be used with the classical metal plasticity models in Abaqus ("Classical metal plasticity," Section 23.2.1) except one-dimensional elements in Abaqus/Explicit (beams and trusses). In Abaqus/Standard it can also be defined for any element that can be used with the linear kinematic hardening plasticity model ("Models for metals subjected to cyclic loading," Section 23.2.2) but not with the nonlinear isotropic/kinematic hardening model. Likewise, anisotropic creep can be defined for any element that can be used with the classical metal creep model in Abaqus/Standard ("Rate-dependent plasticity: creep and swelling," Section 23.2.4).

Output

The standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2) and all output variables associated with the creep model ("Rate-dependent plasticity: creep and swelling," Section 23.2.4), classical metal plasticity models ("Classical metal plasticity," Section 23.2.1), and the linear kinematic hardening plasticity model ("Models for metals subjected to cyclic loading," Section 23.2.2) are available when anisotropic yield and creep are defined.

The following variables have special meaning if anisotropic yield and creep are defined:

PEEQ	Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\|_0 + \int_0^t \dot{\bar{\varepsilon}}^{pl} dt = \bar{\varepsilon}^{pl}\|_0 + \int_0^t \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl} dt}{\sigma^0}$, where $\bar{\varepsilon}^{pl}\|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions").
CEEQ	Equivalent creep strain, $\bar{\varepsilon}^{cr} = \int_0^t \dot{\bar{\varepsilon}}^{cr} dt = \int_0^t \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{cr} dt}{\sigma^0}$.

23.2.7 JOHNSON-COOK PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Classical metal plasticity," Section 23.2.1
- "Rate-dependent yield," Section 23.2.3
- "Equation of state," Section 25.2.1
- Chapter 24, "Progressive Damage and Failure"
- "Dynamic failure models," Section 23.2.8
- "Annealing or melting," Section 23.2.5
- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *ANNEAL TEMPERATURE
- *PLASTIC
- *RATE DEPENDENT
- *SHEAR FAILURE
- *TENSILE FAILURE
- *DAMAGE INITIATION
- *DAMAGE EVOLUTION
- "Using the Johnson-Cook hardening model to define classical metal plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The Johnson-Cook plasticity model:

- is a particular type of Mises plasticity model with analytical forms of the hardening law and rate dependence;
- is suitable for high-strain-rate deformation of many materials, including most metals;
- is typically used in adiabatic transient dynamic simulations;
- can be used in conjunction with the Johnson-Cook dynamic failure model in Abaqus/Explicit;
- can be used in conjunction with the tensile failure model to model tensile spall or a pressure cutoff in Abaqus/Explicit;
- can be used in conjunction with the progressive damage and failure models (Chapter 24, "Progressive Damage and Failure") to specify different damage initiation criteria and damage

evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh; and

- must be used in conjunction with either the linear elastic material model ("Linear elastic behavior," Section 22.2.1) or the equation of state material model ("Equation of state," Section 25.2.1).

Yield surface and flow rule

A Mises yield surface with associated flow is used in the Johnson-Cook plasticity model.

Johnson-Cook hardening

Johnson-Cook hardening is a particular type of isotropic hardening where the static yield stress, σ^0, is assumed to be of the form

$$\sigma^0 = \left[A + B\left(\bar{\varepsilon}^{pl}\right)^n\right]\left(1 - \hat{\theta}^m\right),$$

where $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain and A, B, n and m are material parameters measured at or below the transition temperature, $\theta_{\text{transition}}$. $\hat{\theta}$ is the nondimensional temperature defined as

$$\hat{\theta} \equiv \begin{cases} 0 & \text{for} \quad \theta < \theta_{\text{transition}} \\ (\theta - \theta_{\text{transition}})/(\theta_{\text{melt}} - \theta_{\text{transition}}) & \text{for} \quad \theta_{\text{transition}} \le \theta \le \theta_{\text{melt}} \\ 1 & \text{for} \quad \theta > \theta_{\text{melt}} \end{cases},$$

where θ is the current temperature, θ_{melt} is the melting temperature, and $\theta_{\text{transition}}$ is the transition temperature defined as the one at or below which there is no temperature dependence on the expression of the yield stress. The material parameters must be measured at or below the transition temperature.

When $\theta \ge \theta_{\text{melt}}$, the material will be melted and will behave like a fluid; there will be no shear resistance since $\sigma^0 = 0$. The hardening memory will be removed by setting the equivalent plastic strain to zero. If backstresses are specified for the model, these will also be set to zero.

If you include annealing behavior in the material definition and the annealing temperature is defined to be less than the melting temperature specified for the metal plasticity model, the hardening memory will be removed at the annealing temperature and the melting temperature will be used strictly to define the hardening function. Otherwise, the hardening memory will be removed automatically at the melting temperature. If the temperature of the material point falls below the annealing temperature at a subsequent point in time, the material point can work harden again. For more details, see "Annealing or melting," Section 23.2.5.

You provide the values of A, B, n, m, θ_{melt}, and $\theta_{\text{transition}}$ as part of the metal plasticity material definition.

Input File Usage: *PLASTIC, HARDENING=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Johnson-Cook**

Johnson-Cook strain rate dependence

Johnson-Cook strain rate dependence assumes that

$$\bar{\sigma} = \sigma^0(\bar{\varepsilon}^{pl}, \theta) R(\dot{\bar{\varepsilon}}^{pl})$$

and

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \exp\left[\frac{1}{C}(R-1)\right] \quad \text{for} \quad \bar{\sigma} \geq \sigma^0,$$

where

$\bar{\sigma}$	is the yield stress at nonzero strain rate;
$\dot{\bar{\varepsilon}}^{pl}$	is the equivalent plastic strain rate;
$\dot{\varepsilon}_0$ and C	are material parameters measured at or below the transition temperature, $\theta_{\text{transition}}$;
$\sigma^0(\bar{\varepsilon}^{pl}, \theta)$	is the static yield stress; and
$R(\dot{\bar{\varepsilon}}^{pl})$	is the ratio of the yield stress at nonzero strain rate to the static yield stress (so that $R(\dot{\varepsilon}_0) = 1.0$).

The yield stress is, therefore, expressed as

$$\bar{\sigma} = \left[A + B\left(\bar{\varepsilon}^{pl}\right)^n\right]\left[1 + C \ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right)\right]\left(1 - \hat{\theta}^m\right).$$

You provide the values of C and $\dot{\varepsilon}_0$ when you define Johnson-Cook rate dependence.

The use of Johnson-Cook hardening does not necessarily require the use of Johnson-Cook strain rate dependence.

Input File Usage: Use both of the following options:

```
*PLASTIC, HARDENING=JOHNSON COOK
*RATE DEPENDENT, TYPE=JOHNSON COOK
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Hardening: Johnson-Cook**: **Suboptions→Rate Dependent**: **Hardening: Johnson-Cook**

Johnson-Cook dynamic failure

Abaqus/Explicit provides a dynamic failure model specifically for the Johnson-Cook plasticity model, which is suitable only for high-strain-rate deformation of metals. This model is referred to as the "Johnson-Cook dynamic failure model." Abaqus/Explicit also offers a more general implementation of the Johnson-Cook failure model as part of the family of damage initiation criteria, which is the recommended technique for modeling progressive damage and failure of materials (see "Damage and

failure for ductile metals: overview," Section 24.2.1). The Johnson-Cook dynamic failure model is based on the value of the equivalent plastic strain at element integration points; failure is assumed to occur when the damage parameter exceeds 1. The damage parameter, ω, is defined as

$$\omega = \sum \left(\frac{\Delta \bar{\varepsilon}^{pl}}{\bar{\varepsilon}_f^{pl}} \right),$$

where $\Delta \bar{\varepsilon}^{pl}$ is an increment of the equivalent plastic strain, $\bar{\varepsilon}_f^{pl}$ is the strain at failure, and the summation is performed over all increments in the analysis. The strain at failure, $\bar{\varepsilon}_f^{pl}$, is assumed to be dependent on a nondimensional plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}/\dot{\varepsilon}_0$; a dimensionless pressure-deviatoric stress ratio, p/q (where $\boldsymbol{p}$ is the pressure stress and $\boldsymbol{q}$ is the Mises stress); and the nondimensional temperature, $\hat{\theta}$, defined earlier in the Johnson-Cook hardening model. The dependencies are assumed to be separable and are of the form

$$\bar{\varepsilon}_f^{pl} = \left[d_1 + d_2 \exp\left(d_3 \frac{p}{q} \right) \right] \left[1 + d_4 \ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0} \right) \right] \left(1 + d_5 \hat{\theta} \right),$$

where d_1–d_5 are failure parameters measured at or below the transition temperature, $\theta_{\text{transition}}$, and $\dot{\varepsilon}_0$ is the reference strain rate. You provide the values of d_1–d_5 when you define the Johnson-Cook dynamic failure model. This expression for $\bar{\varepsilon}_f^{pl}$ differs from the original formula published by Johnson and Cook (1985) in the sign of the parameter d_3. This difference is motivated by the fact that most materials experience an increase in $\bar{\varepsilon}_f^{pl}$ with increasing pressure-deviatoric stress ratio; therefore, d_3 in the above expression will usually take positive values.

When this failure criterion is met, the deviatoric stress components are set to zero and remain zero for the rest of the analysis. Depending on your choice, the pressure stress may also be set to zero for the rest of calculation (if this is the case, you must specify element deletion and the element will be deleted) or it may be required to remain compressive for the rest of the calculation (if this is the case, you must choose not to use element deletion). By default, the elements that meet the failure criterion are deleted.

The Johnson-Cook dynamic failure model is suitable for high-strain-rate deformation of metals; therefore, it is most applicable to truly dynamic situations. For quasi-static problems that require element removal, the progressive damage and failure models (Chapter 24, "Progressive Damage and Failure") or the Gurson metal plasticity model ("Porous metal plasticity," Section 23.2.9) are recommended.

The use of the Johnson-Cook dynamic failure model requires the use of Johnson-Cook hardening but does not necessarily require the use of Johnson-Cook strain rate dependence. However, the rate-dependent term in the Johnson-Cook dynamic failure criterion will be included only if Johnson-Cook strain rate dependence is defined. The Johnson-Cook damage initiation criterion described in "Damage initiation for ductile metals," Section 24.2.2, does not have these limitations.

Input File Usage: Use both of the following options:

```
*PLASTIC, HARDENING=JOHNSON COOK
*SHEAR FAILURE, TYPE=JOHNSON COOK,
ELEMENT DELETION=YES or NO
```

Abaqus/CAE Usage: Johnson-Cook dynamic failure is not supported in Abaqus/CAE.

Progressive damage and failure

The Johnson-Cook plasticity model can be used in conjunction with the progressive damage and failure models discussed in "Damage and failure for ductile metals: overview," Section 24.2.1. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), Müschenborn-Sonne forming limit diagram (MSFLD), and, in Abaqus/Explicit, Marciniak-Kuczynski (M-K) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The models offer two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations. This is a great advantage over the dynamic failure models discussed above.

Input File Usage: Use the following options:

```
*PLASTIC, HARDENING=JOHNSON COOK
*DAMAGE INITIATION
*DAMAGE EVOLUTION
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*damage initiation type***: specify the damage initiation criterion: **Suboptions→Damage Evolution**: specify the damage evolution parameters

Tensile failure

In Abaqus/Explicit the tensile failure model can be used in conjunction with the Johnson-Cook plasticity model to define tensile failure of the material. The tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff and offers a number of failure choices including element removal. Similar to the Johnson-Cook dynamic failure model, the Abaqus/Explicit tensile failure model is suitable for high-strain-rate deformation of metals and is most applicable to truly dynamic problems. For more details, see "Dynamic failure models," Section 23.2.8.

Input File Usage: Use both of the following options:

```
*PLASTIC, HARDENING=JOHNSON COOK
*TENSILE FAILURE
```

Abaqus/CAE Usage: The tensile failure model is not supported in Abaqus/CAE.

Heat generation by plastic work

Abaqus allows for an adiabatic thermal-stress analysis ("Adiabatic analysis," Section 6.5.4), a fully coupled temperature-displacement analysis ("Fully coupled thermal-stress analysis," Section 6.5.3), or a fully coupled thermal-electrical-structural analysis ("Fully coupled thermal-electrical-structural analysis," Section 6.7.4) to be performed in which heat generated by plastic straining of a material is calculated. This method is typically used in the simulation of bulk metal forming or high-speed

manufacturing processes involving large amounts of inelastic strain, where the heating of the material caused by its deformation is an important effect because of temperature dependence of the material properties. Since the Johnson-Cook plasticity model is motivated by high-strain-rate transient dynamic applications, temperature change in this model is generally computed by assuming adiabatic conditions (no heat transfer between elements). Heat is generated in an element by plastic work, and the resulting temperature rise is computed using the specific heat of the material.

This effect is introduced by defining the fraction of the rate of inelastic dissipation that appears as a heat flux per volume.

Input File Usage: Use all of the following options in the same material data block:

*PLASTIC, HARDENING=JOHNSON COOK
*SPECIFIC HEAT
*DENSITY
*INELASTIC HEAT FRACTION

Abaqus/CAE Usage: Use all of the following options in the same material definition:

Property module: material editor:
Mechanical→Plasticity→Plastic: **Hardening: Johnson-Cook**
Thermal→Specific Heat
General→Density
Thermal→Inelastic Heat Fraction

Initial conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, initial equivalent plastic strain values can be provided to specify the yield stress corresponding to the work hardened state (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). An initial backstress, α_0, can also be specified. The backstress α_0 represents a constant kinematic shift of the yield surface, which can be useful for modeling the effects of residual stresses without considering them in the equilibrium solution.

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Elements

The Johnson-Cook plasticity model can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning for the Johnson-Cook plasticity model:

PEEQ	Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}|_0 + \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\bar{\varepsilon}^{pl}|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions").
STATUS	Status of element. The status of an element is 1.0 if the element is active and 0.0 if the element is not.

Additional reference

- Johnson, G. R., and W. H. Cook, "Fracture Characteristics of Three Metals Subjected to Various Strains, Strain rates, Temperatures and Pressures," Engineering Fracture Mechanics, vol. 21, no. 1, pp. 31–48, 1985.

23.2.8 DYNAMIC FAILURE MODELS

Product: Abaqus/Explicit

References

- "Equation of state," Section 25.2.1
- "Classical metal plasticity," Section 23.2.1
- "Rate-dependent yield," Section 23.2.3
- "Johnson-Cook plasticity," Section 23.2.7
- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *SHEAR FAILURE
- *TENSILE FAILURE

Overview

The progressive damage and failure models described in "Damage and failure for ductile metals: overview," Section 24.2.1, are the recommended method for modeling material damage and failure in Abaqus; these models are suitable for both quasi-static and dynamic situations. Abaqus/Explicit offers two additional element failure models suitable only for high-strain-rate dynamic problems. The shear failure model is driven by plastic yielding. The tensile failure model is driven by tensile loading. These failure models can be used to limit subsequent load-carrying capacity of an element (up to the point of removing the element) once a stress limit is reached. Both models can be used for the same material.

The shear failure model:

- is designed for high-strain-rate deformation of many materials, including most metals;
- uses the equivalent plastic strain as a failure measure;
- offers two choices for what occurs upon failure, including the removal of elements from the mesh;
- can be used in conjunction with either the Mises or the Johnson-Cook plasticity models; and
- can be used in conjunction with the tensile failure model.

The tensile failure model:

- is designed for high-strain-rate deformation of many materials, including most metals;
- uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff;
- offers a number of choices for what occurs upon failure, including the removal of elements from the mesh;
- can be used in conjunction with either the Mises or the Johnson-Cook plasticity models or the equation of state material model; and
- can be used in conjunction with the shear failure model.

Shear failure model

The shear failure model can be used in conjunction with the Mises or the Johnson-Cook plasticity models in Abaqus/Explicit to define shear failure of the material.

Shear failure criterion

The shear failure model is based on the value of the equivalent plastic strain at element integration points; failure is assumed to occur when the damage parameter exceeds 1. The damage parameter, ω, is defined as

$$\omega = \frac{\bar{\varepsilon}_0^{pl} + \sum \Delta\bar{\varepsilon}^{pl}}{\bar{\varepsilon}_f^{pl}},$$

where $\bar{\varepsilon}_0^{pl}$ is any initial value of the equivalent plastic strain, $\Delta\bar{\varepsilon}^{pl}$ is an increment of the equivalent plastic strain, $\bar{\varepsilon}_f^{pl}$ is the strain at failure, and the summation is performed over all increments in the analysis.

The strain at failure, $\bar{\varepsilon}_f^{pl}$, is assumed to depend on the plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$; a dimensionless pressure-deviatoric stress ratio, p/q (where $\boldsymbol{p}$ is the pressure stress and $\boldsymbol{q}$ is the Mises stress); temperature; and predefined field variables. There are two ways to define the strain at failure, $\bar{\varepsilon}_f^{pl}$. One is to use direct tabular data, where the dependencies are given in a tabular form. Alternatively, the analytical form proposed by Johnson and Cook can be invoked (see "Johnson-Cook plasticity," Section 23.2.7, for more details).

When direct tabular data are used to define the shear failure model, the strain at failure, $\bar{\varepsilon}_f^{pl}$, must be given as a tabular function of the equivalent plastic strain rate, the pressure-deviatoric stress ratio, temperature, and predefined field variables. This method requires the use of the Mises plasticity model.

For the Johnson-Cook shear failure model, you must specify the failure parameters, d_1–d_5 (see "Johnson-Cook plasticity," Section 23.2.7, for more details on these parameters). The shear failure data must be calibrated at or below the transition temperature, $\theta_{\text{transition}}$, defined in "Johnson-Cook plasticity," Section 23.2.7. This method requires the use of the Johnson-Cook plasticity model.

Input File Usage: Use both of the following options for the Mises plasticity model:

```
*PLASTIC, HARDENING=ISOTROPIC
*SHEAR FAILURE, TYPE=TABULAR
```

Use both of the following options for the Johnson-Cook plasticity model:

```
*PLASTIC, HARDENING=JOHNSON COOK
*SHEAR FAILURE, TYPE=JOHNSON COOK
```

Element removal

When the shear failure criterion is met at an integration point, all the stress components will be set to zero and that material point fails. By default, if all of the material points at any one section of an element fail, the element is removed from the mesh; it is not necessary for all material points in the element to fail. For example, in a first-order reduced-integration solid element removal of the element takes place as soon as

its only integration point fails. However, in a shell element all through-the-thickness integration points must fail before the element is removed from the mesh. In the case of second-order reduced-integration beam elements, failure of all integration points through the section at either of the two element integration locations along the beam axis leads, by default, to element removal. Similarly, in the modified triangular and tetrahedral solid elements failure at any one integration point leads, by default, to element removal. Element deletion is the default failure choice.

An alternative failure choice, where the element is not deleted, is to specify that when the shear failure criterion is met at a material point, the deviatoric stress components will be set to zero for that point and will remain zero for the rest of the calculation. The pressure stress is then required to remain compressive; that is, if a negative pressure stress is computed in a failed material point in an increment, it is reset to zero. This failure choice is not allowed when using plane stress, shell, membrane, beam, pipe, and truss elements because the structural constraints may be violated.

Input File Usage: Use the following option to allow element deletion when the failure criterion is met (the default):

*SHEAR FAILURE, ELEMENT DELETION=YES

Use the following option to allow the element to take hydrostatic compressive stress only when the failure criterion is met:

*SHEAR FAILURE, ELEMENT DELETION=NO

Determining when to use the shear failure model

The shear failure model in Abaqus/Explicit is suitable for high-strain-rate dynamic problems where inertia is important. Improper use of the shear failure model may result in an incorrect simulation.

For quasi-static problems that may require element removal, the progressive damage and failure models (Chapter 24, "Progressive Damage and Failure") or the Gurson porous metal plasticity model ("Porous metal plasticity," Section 23.2.9) are recommended.

Tensile failure model

The tensile failure model can be used in conjunction with either the Mises or the Johnson-Cook plasticity models or the equation of state material model in Abaqus/Explicit to define tensile failure of the material.

Tensile failure criterion

The Abaqus/Explicit tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff. The tensile failure criterion assumes that failure occurs when the pressure stress, p, becomes more tensile than the user-specified hydrostatic cutoff stress, σ_{cutoff}. The hydrostatic cutoff stress may be a function of temperature and predefined field variables. There is no default value for this stress.

The tensile failure model can be used with either the Mises or the Johnson-Cook plasticity models or the equation of state material model.

Input File Usage: Use both of the following options for the Mises or Johnson-Cook plasticity models:

∗PLASTIC
∗TENSILE FAILURE

Use both of the following options for the equation of state material model:

∗EOS
∗TENSILE FAILURE

Failure choices

When the tensile failure criterion is met at an element integration point, the material point fails. Five failure choices are offered for the failed material points: the default choice, which includes element removal, and four different spall models. These failure choices are described below.

Element removal

When the tensile failure criterion is met at an integration point, all the stress components will be set to zero and that material point fails. By default, if all of the material points at any one section of an element fail, the element is removed from the mesh; it is not necessary for all material points in the element to fail. For example, in a first-order reduced-integration solid element removal of the element takes place as soon as its only integration point fails. However, in a shell element all through-the-thickness integration points must fail before the element is removed from the mesh. In the case of second-order reduced-integration beam elements, failure of all integration points through the section at either of the two element integration locations along the beam axis leads, by default, to element removal. Similarly, in the modified triangular and tetrahedral solid elements failure at any one integration point leads, by default, to element removal.

Input File Usage: ∗TENSILE FAILURE, ELEMENT DELETION=YES (default)

Spall models

An alternative failure choice that is based on spall (the crumbling of a material), rather than element removal, is also available. Four failure combinations are available in this category. When the tensile failure criterion is met at a material point, the deviatoric stress components may be unaffected or may be required to be zero, and the pressure stress may be limited by the hydrostatic cutoff stress or may be required to be compressive. Therefore, there are four possible failure combinations (see Figure 23.2.8–1, where "O" is the stress that would exist if the tensile failure model were not used). These failure combinations are as follows:

- Ductile shear and ductile pressure: this choice corresponds to point 1 in Figure 23.2.8–1 and models the case in which the deviatoric stress components are unaffected and the pressure stress is limited by the hydrostatic cutoff stress; i.e., $p = max(-\sigma_{\text{cutoff}}, p)$.

  Input File Usage: ∗TENSILE FAILURE, ELEMENT DELETION=NO,
  SHEAR=DUCTILE, PRESSURE=DUCTILE

- Brittle shear and ductile pressure: this choice corresponds to point 2 in Figure 23.2.8–1 and models the case in which the deviatoric stress components are set to zero and remain zero for

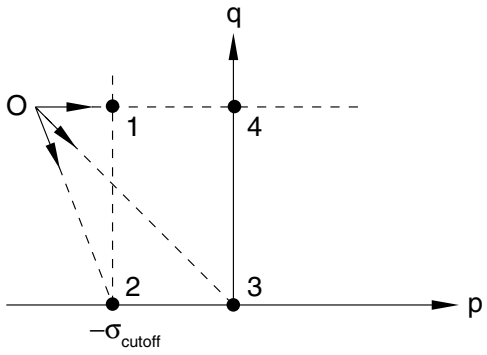


Figure 23.2.8–1 Tensile failure choices.

the rest of the calculation, and the pressure stress is limited by the hydrostatic cutoff stress; i.e., $p = max(-\sigma_{\text{cutoff}}, p)$.

Input File Usage: ∗TENSILE FAILURE, ELEMENT DELETION=NO,
SHEAR=BRITTLE, PRESSURE=DUCTILE

- Brittle shear and brittle pressure: this choice corresponds to point 3 in Figure 23.2.8–1 and models the case in which the deviatoric stress components are set to zero and remain zero for the rest of the calculation, and the pressure stress is required to be compressive; i.e., $p = max(0, p)$.

  Input File Usage: ∗TENSILE FAILURE, ELEMENT DELETION=NO,
  SHEAR=BRITTLE, PRESSURE=BRITTLE

- Ductile shear and brittle pressure: this choice corresponds to point 4 in Figure 23.2.8–1 and models the case in which the deviatoric stress components are unaffected and the pressure stress is required to be compressive; i.e., $p = max(0, p)$.

  Input File Usage: ∗TENSILE FAILURE, ELEMENT DELETION=NO,
  SHEAR=DUCTILE, PRESSURE=BRITTLE

There is no default failure combination for the spall models. If you choose not to use the element deletion model, you must specify the failure combination explicitly. If the material's deviatoric behavior is not defined (for example, the equation of state model without deviatoric behavior is used), the deviatoric part of the combination is meaningless and will be ignored. The spall models are not allowed when using plane stress, shell, membrane, beam, pipe, and truss elements.

Determining when to use the tensile failure model

The tensile failure model in Abaqus/Explicit is suitable for high-strain-rate dynamic problems in which inertia effects are important. Improper use of the tensile failure model may result in an incorrect simulation.

Using the failure models with rebar

It is possible to use the shear failure and/or the tensile failure models in elements for which rebars are also defined. When such elements fail according to the failure criterion, the base material contribution to the element stress-carrying capacity is removed or adjusted depending on the type of failure chosen, but the rebar contribution to the element stress-carrying capacity is not removed. However, if you also include failure in the rebar material definition, the rebar contribution to the element stress-carrying capacity will also be removed or adjusted if the failure criterion specified for the rebar is met.

Elements

The shear and tensile failure models with element deletion can be used with any elements in Abaqus/Explicit that include mechanical behavior (elements that have displacement degrees of freedom). The shear and tensile failure models without element deletion can be used only with plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus/Explicit.

Output

In addition to the standard output identifiers available in Abaqus/Explicit ("Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variable has special meaning for the shear and tensile failure models:

STATUS	Status of element (the status of an element is 1.0 if the element is active, 0.0 if the element is not).

23.2.9 POROUS METAL PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *POROUS METAL PLASTICITY
- *POROUS FAILURE CRITERIA
- *VOID NUCLEATION
- "Defining porous metal plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The porous metal plasticity model:

- is used to model materials with a dilute concentration of voids in which the relative density is greater than 0.9;
- is based on Gurson's porous metal plasticity theory (Gurson, 1977) with void nucleation and, in Abaqus/Explicit, a failure definition; and
- defines the inelastic flow of the porous metal on the basis of a potential function that characterizes the porosity in terms of a single state variable, the relative density.

Elastic and plastic behavior

You specify the elastic part of the response separately; only linear isotropic elasticity can be specified (see "Linear elastic behavior," Section 22.2.1).

You specify the hardening behavior of the fully dense matrix material by defining a metal plasticity model (see "Classical metal plasticity," Section 23.2.1). Only isotropic hardening can be specified. The hardening curve must describe the yield stress of the matrix material as a function of plastic strain in the matrix material. In defining this dependence at finite strains, "true" (Cauchy) stress and log strain values should be given. Rate dependency effects for the matrix material can be modeled (see "Rate-dependent yield," Section 23.2.3).

Yield condition

The relative density of a material, r, is defined as the ratio of the volume of solid material to the total volume of the material. The relationships defining the model are expressed in terms of the void volume fraction, f, which is defined as the ratio of the volume of voids to the total volume of the material. It follows that $f = 1 - r$. For a metal containing a dilute concentration of voids, Gurson (1977) proposed

a yield condition as a function of the void volume fraction. This yield condition was later modified by Tvergaard (1981) to the form

$$\Phi = \left(\frac{q}{\sigma_y}\right)^2 + 2q_1 f \cosh\left(-q_2\frac{3p}{2\sigma_y}\right) - \left(1 + q_3 f^2\right) = 0,$$

where

$\mathbf{S} = p\mathbf{I} + \sigma$	is the deviatoric part of the Cauchy stress tensor $\boldsymbol{\sigma}$;
$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$	is the effective Mises stress;
$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$	is the hydrostatic pressure;
$\sigma_y(\bar{\varepsilon}_m^{pl})$	is the yield stress of the fully dense matrix material as a function of $\bar{\varepsilon}_m^{pl}$, the equivalent plastic strain in the matrix; and
q_1, q_2, q_3	are material parameters.

The Cauchy stress is defined as the force per "current unit area," comprised of voids and the solid (matrix) material.

$f = 0$ ($r = 1$) implies that the material is fully dense, and the Gurson yield condition reduces to the Mises yield condition. $f = 1$ ($r = 0$) implies that the material is completely voided and has no stress carrying capacity. The model generally gives physically reasonable results only for $f < 0.1$ ($r > 0.9$).

The model is described in detail in "Porous metal plasticity," Section 4.3.6 of the Abaqus Theory Manual, along with a discussion of its numerical implementation.

If the porous metal plasticity model is used during a pore pressure analysis (see "Coupled pore fluid diffusion and stress analysis," Section 6.8.1), the relative density, r, is tracked independently of the void ratio.

Specifying q_1, q_2, and q_3

You specify the parameters q_1, q_2, and q_3 directly for the porous metal plasticity model. For typical metals the ranges of the parameters reported in the literature are q_1 = 1.0 to 1.5, q_2 = 1.0, and $q_3 = q_1^2$ = 1.0 to 2.25 (see "Necking of a round tensile bar," Section 1.1.9 of the Abaqus Benchmarks Manual). The original Gurson model is recovered when $q_1 = q_2 = q_3 = 1.0$. You can define these parameters as tabular functions of temperature and/or field variables.

Input File Usage: *POROUS METAL PLASTICITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Porous Metal Plasticity**

Failure criteria in Abaqus/Explicit

The porous metal plasticity model in Abaqus/Explicit allows for failure. In this case the yield condition is written as

$$\Phi = \left(\frac{q}{\sigma_y}\right)^2 + 2q_1 f^* \cosh\left(-q_2\frac{3p}{2\sigma_y}\right) - \left(1 + q_3 f^{*2}\right) = 0,$$

where the function $f^*(f)$ models the rapid loss of stress carrying capacity that accompanies void coalescence. This function is defined in terms of the void volume fraction:

$$f^* = \begin{cases} f & \text{if } f \le f_c, \\ f_c + \dfrac{\bar{f}_F - f_c}{f_F - f_c}(f - f_c) & \text{if } f_c < f < f_F \\ \bar{f}_F & \text{if } f \ge f_F, \end{cases},$$

where

$$\bar{f}_F = \frac{q_1 + \sqrt{q_1^2 - q_3}}{q_3}.$$

In the above relationship f_c is a critical value of the void volume fraction, and f_F is the value of void volume fraction at which there is a complete loss of stress carrying capacity in the material. The user-specified parameters f_c and f_F model the material failure when $f_c < f < f_F$, due to mechanisms such as micro fracture and void coalescence. When $f \ge f_F$, total failure at the material point occurs. In Abaqus/Explicit an element is removed once all of its material points have failed.

Input File Usage: Use the following option in conjunction with the *POROUS METAL PLASTICITY option:

*POROUS FAILURE CRITERIA

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Porous Metal Plasticity**: **Suboptions→Porous Failure Criteria**

Specifying the initial relative density

You can specify the initial relative density of the porous material, r_0, at material points or at nodes. If you do not specify the initial relative density, Abaqus will assign it a value of 1.0.

At material points

You can specify the initial relative density as part of the porous metal plasticity material definition.

Input File Usage: *POROUS METAL PLASTICITY, RELATIVE DENSITY=r_0

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Porous Metal Plasticity**: **Relative density:** r_0

At nodes

Alternatively, you can specify the initial relative density at nodes as initial conditions (“Initial conditions in Abaqus/Standard and Abaqus/Explicit,” Section 33.2.1); these values are interpolated to the material points. The initial conditions are applied only if the relative density is not specified as part of the porous metal plasticity material definition. When a discontinuity of the initial relative density field occurs at the element boundaries, separate nodes must be used to define the elements at these boundaries, with multi-point constraints applied to make the nodal displacements and rotations equivalent.

Input File Usage: *INITIAL CONDITIONS, TYPE=RELATIVE DENSITY

Abaqus/CAE Usage: Initial relative density is not supported in Abaqus/CAE.

Flow rule and hardening

The presence of pressure in the yield condition results in nondeviatoric plastic strains. Plastic flow is assumed to be normal to the yield surface:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda}\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}.$$

The hardening of the fully dense matrix material is described through $\sigma_y = \sigma_y(\bar{\varepsilon}_m^{pl})$. The evolution of the equivalent plastic strain in the matrix material is obtained from the following equivalent plastic work expression:

$$(1-f)\sigma_y \dot{\bar{\varepsilon}}_m^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}.$$

The model is illustrated in Figure 23.2.9–1, where the yield surfaces for different levels of void volume fraction are shown in the p–q plane.

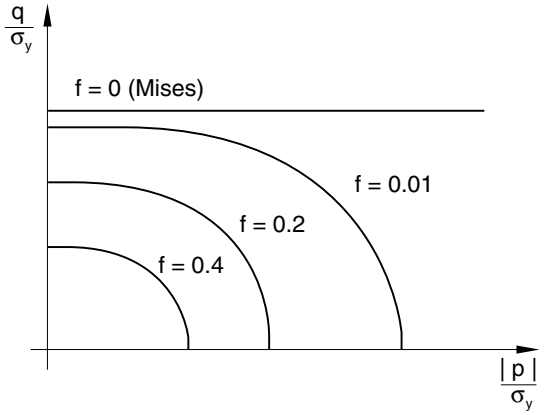


Figure 23.2.9–1 Schematic of the yield surface in the p–q plane.

Figure 23.2.9–2 compares the behavior of a porous material (whose initial yield stress is σ_{y_0}) in tension and compression against the behavior of the perfectly plastic matrix material. In compression the porous material "hardens" due to closing of the voids, and in tension it "softens" due to growth and nucleation of the voids.

Void growth and nucleation

The total change in void volume fraction is given as

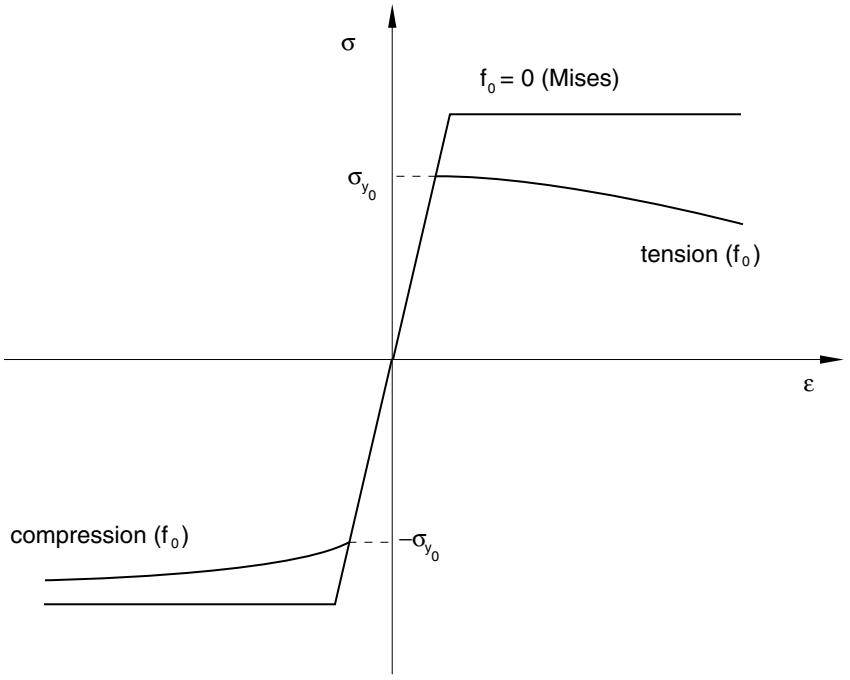


Figure 23.2.9–2 Schematic of uniaxial behavior of a porous metal (perfectly plastic matrix material with initial volume fraction of voids = f_0).

$$\dot{f} = \dot{f}_{\text{gr}} + \dot{f}_{\text{nucl}},$$

where $\dot{f}_{\text{gr}}$ is change due to growth of existing voids and $\dot{f}_{\text{nucl}}$ is change due to nucleation of new voids. Growth of the existing voids is based on the law of conservation of mass and is expressed in terms of the void volume fraction:

$$\dot{f}_{\text{gr}} = (1 - f)\,\dot{\boldsymbol{\varepsilon}}^{pl} : \mathbf{I}.$$

The nucleation of voids is given by a strain-controlled relationship:

$$\dot{f}_{\text{nucl}} = A\,\dot{\bar{\varepsilon}}^{pl}_{m},$$

where

$$A = \frac{f_N}{s_N\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\bar{\varepsilon}^{pl}_{m} - \varepsilon_N}{s_N}\right)^2\right].$$

The normal distribution of the nucleation strain has a mean value ε_N and standard deviation s_N. f_N is the volume fraction of the nucleated voids, and voids are nucleated only in tension.

The nucleation function A/f_N is assumed to have a normal distribution, as shown in Figure 23.2.9–3 for different values of the standard deviation s_N.

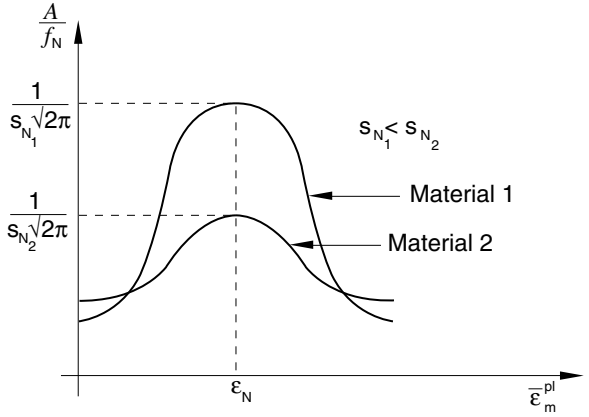


Figure 23.2.9–3 Nucleation function A/f_N.

Figure 23.2.9–4 shows the extent of softening in a uniaxial tension test of a porous material for different values of f_N.

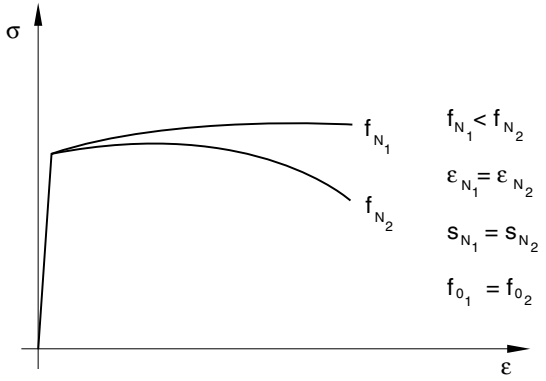


Figure 23.2.9–4 Softening (in uniaxial tension) as a function of f_N.

The following ranges of values are reported in the literature for typical metals: $\varepsilon_N = 0.1$ to 0.3, $s_N =$ 0.05 to 0.1, and $f_N = 0.04$ (see "Necking of a round tensile bar," Section 1.1.9 of the Abaqus Benchmarks Manual). You specify these parameters, which can be defined as tabular functions of temperature and

predefined field variables. Abaqus will include void nucleation in a tensile field only when you include it in the material definition.

In Abaqus/Standard the accuracy of the implicit integration of the void nucleation and growth equation is controlled by prescribing the maximum allowable time increment in the automatic time incrementation scheme.

Input File Usage: *VOID NUCLEATION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Porous Metal Plasticity**: **Suboptions→Void Nucleation**

Initial conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, Abaqus allows you to prescribe initial conditions directly for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Defining initial hardening conditions in a user subroutine

For more complicated cases, initial conditions can be defined in Abaqus/Standard through user subroutine **HARDINI**.

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING, USER

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

The porous metal plasticity model can be used with any stress/displacement elements other than one-dimensional elements (beam, pipe, and truss elements) or elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning in the porous metal plasticity model:

PEEQ	Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\|_0 + \int \frac{\boldsymbol{\sigma}:d\boldsymbol{\varepsilon}^{pl}}{(1-f)\sigma_y}$, where $\bar{\varepsilon}^{pl}\|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions").
VVF	Void volume fraction.

VVFG	Void volume fraction due to void growth.
VVFN	Void volume fraction due to void nucleation.

Additional references

- Gurson, A. L., "Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Materials," Journal of Engineering Materials and Technology, vol. 99, pp. 2–15, 1977.
- Tvergaard, V., "Influence of Voids on Shear Band Instabilities under Plane Strain Condition," International Journal of Fracture Mechanics, vol. 17, pp. 389–407, 1981.

23.2.10 CAST IRON PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Combining material behaviors," Section 21.1.3
- "Inelastic behavior," Section 23.1.1
- *CAST IRON COMPRESSION HARDENING
- *CAST IRON PLASTICITY
- *CAST IRON TENSION HARDENING
- "Defining cast iron plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The cast iron plasticity model:

- is intended for the constitutive modeling of gray cast iron;
- provides elastic-plastic behavior with different yield strengths, flow, and hardening in tension and compression;
- is based on a yield function that depends on the maximum principal stress under tensile loading conditions and pressure-independent (von Mises type) behavior under compressive loading conditions;
- allows for simultaneous inelastic dilatation and inelastic shearing under tensile loading conditions;
- allows only inelastic shearing under compressive loading conditions;
- is intended for the simulation of material response only under essentially monotonic loading conditions; and
- cannot be used to model rate dependence.

Elastic and plastic behavior

The cast iron plasticity model describes the mechanical behavior of gray cast iron, a material with a microstructure consisting of a distribution of graphite flakes in a steel matrix. In tension the graphite flakes act as stress concentrators, resulting in yielding as a function of the maximum principal stress, followed by brittle behavior. In compression the graphite flakes do not have an appreciable effect on the macroscopic response, resulting in a ductile behavior similar to that of many steels.

You specify the elastic part of the response separately; only linear isotropic elasticity can be used (see "Linear elastic behavior," Section 22.2.1). The elastic stiffness is assumed to be the same under tension and compression.

The cast iron plasticity model is used to provide the value of the plastic "Poisson's ratio," which is the absolute value of the ratio of the transverse to the longitudinal plastic strain under uniaxial tension. The plastic Poisson's ratio can vary with the plastic deformation. However, the model in Abaqus assumes that it is constant with respect to plastic deformation. It can depend on temperature and field variables. If no value is specified for the plastic Poisson's ratio, a default value of 0.04 is assumed. This default value is based on experimental results for permanent volumetric strain under uniaxial tension (see "Cast iron plasticity," Section 4.3.7 of the Abaqus Theory Manual, for details).

Independent hardening (see Figure 23.2.10–1) of the material under tension and compression can be specified as described below. The tension hardening data provide the uniaxial tension yield stress as a function of plastic strain, temperature, and field variables under uniaxial tension. The compression hardening data provide the uniaxial compression yield stress as a function of plastic strain, temperature, and field variables under uniaxial compression.

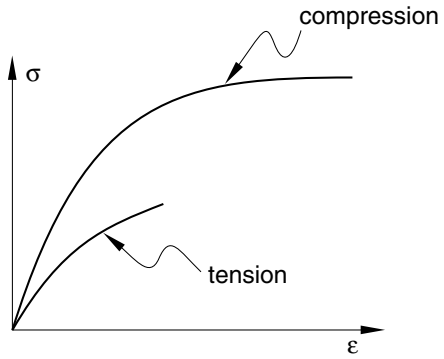


Figure 23.2.10–1 Typical stress-strain response of gray cast iron under uniaxial tension and uniaxial compression.

Input File Usage: *CAST IRON PLASTICITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Cast Iron Plasticity**

Yield condition

Abaqus makes use of a composite yield surface to describe the different behavior in tension and compression. In tension yielding is assumed to be governed by the maximum principal stress, while in compression yielding is assumed to be pressure independent and governed by the deviatoric stresses alone (Mises yield condition).

The model is described in detail in "Cast iron plasticity," Section 4.3.7 of the Abaqus Theory Manual.

Flow rule

For the purposes of discussing the flow and hardening behavior, it is useful to divide the meridional plane into the two regions shown in Figure 23.2.10–2.

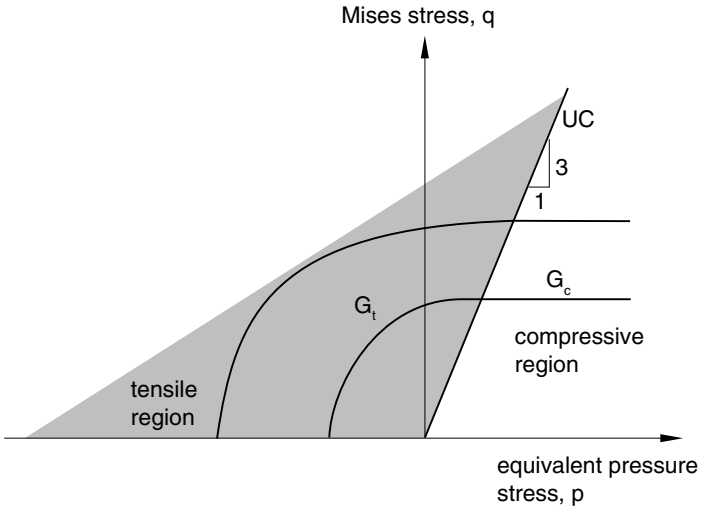


Figure 23.2.10–2 Schematic of the flow potentials in the p–q plane.

The region to the left of the uniaxial compression line (labeled UC) is referred to as the "tensile region," while the region to the right of the uniaxial compression line is referred to as the "compressive region." The flow potential consists of the Mises cylinder in the compressive region and an ellipsoidal "cap" in the tensile region. The transition between the two surfaces is smooth. The projection of the flow potential on the meridional plane (see Figure 23.2.10–2) consists of a straight line in the compressive region and an ellipse in the tensile region. The corresponding projection on the deviatoric plane is a circle. A consequence of the above choice is that plastic flow results in inelastic volume expansion in the tensile region and no inelastic volume change in the compressive region (see "Cast iron plasticity," Section 4.3.7 of the Abaqus Theory Manual, for details).

Nonassociated flow

Since the flow potential is different from the yield surface ("nonassociated" flow), the material Jacobian matrix is unsymmetric. Hence, to improve convergence, use the unsymmetric matrix storage and solution scheme (see "Defining an analysis," Section 6.1.2).

Hardening

Since the hardening of gray cast iron is different in uniaxial tension and uniaxial compression, you need to provide two sets of hardening data in tabular form: one based on a uniaxial tension experiment

that defines $\sigma_t = \sigma_t(\bar{\varepsilon}_t^{pl}, \theta, f^\alpha)$ and the other based on a uniaxial compression experiment that defines $\sigma_c = \sigma_c(\bar{\varepsilon}_c^{pl}, \theta, f^\alpha)$. Here, $\bar{\varepsilon}_t^{pl}$ and $\bar{\varepsilon}_c^{pl}$ are the equivalent plastic strains in uniaxial tension and uniaxial compression, respectively.

Input File Usage: Use both of the following options in conjunction with the *CAST IRON PLASTICITY option:

*CAST IRON COMPRESSION HARDENING
*CAST IRON TENSION HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Cast Iron Plasticity**: **Compression Hardening** and **Tension Hardening**

Restrictions on material data

The plastic Poisson's ratio, ν_{pl}, is expected to be less than 0.5 since experimental results suggest that there is a permanent increase in the volume of gray cast iron when it is loaded in uniaxial tension beyond yield. For the potential to be well-defined, ν_{pl} must be greater than -1.0. Thus, the plastic Poisson's ratio must satisfy $-1.0 < \nu_{pl} \leq 0.5$.

The cast iron plasticity material model is intended for modeling cast iron and other materials like cast iron for which the behavior in uniaxial tension and uniaxial compression matches the behavior shown in Figure 23.2.10–1. In particular, the model expects the initial yield stress in uniaxial tension to be less than the initial yield stress in uniaxial compression. Even if the overall stress-strain response and hardening behavior in uniaxial stress states of some material other than cast iron is consistent with that of cast iron, you must also ensure that the flow potential (which has been constructed specifically for modeling cast iron) for the model is meaningful for other materials. Abaqus issues a warning message only if the initial yield stress in uniaxial tension is equal to or greater than that in uniaxial compression. No other checks are carried out in this regard.

If the yield stress in uniaxial tension is higher than that in uniaxial compression, a material point in uniaxial tension may actually yield at the initial yield stress specified for uniaxial compression. This apparent anomalous behavior is due to the fact that (as a result of unrealistic user-specified material properties) a uniaxial tension stressing path in stress space meets the compressive (Mises) part of the yield surface first.

Elements

The cast iron plasticity model can be used with any stress/displacement element in Abaqus other than elements for which the assumed stress state is plane stress (plane stress continuum, shell, and membrane elements). It can be used with one-dimensional elements (trusses and beams in a plane) and, in Abaqus/Standard, with beams in space.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning for the cast iron plasticity material model:

PEEQ	Equivalent plastic strain in uniaxial compression, $\bar{\varepsilon}_c^{pl} = \int_0^t \dot{\bar{\varepsilon}}_c^{pl}\, dt$.
PEEQT	Equivalent plastic strain in uniaxial tension, $\bar{\varepsilon}_t^{pl} = \int_0^t \dot{\bar{\varepsilon}}_t^{pl}\, dt$.

23.2.11 TWO-LAYER VISCOPLASTICITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Combining material behaviors," Section 21.1.3
- "Inelastic behavior," Section 23.1.1
- *ELASTIC
- *PLASTIC
- *VISCOUS
- "Defining the viscous component of a two-layer viscoplasticity model" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The two-layer viscoplastic model:

- is intended for modeling materials in which significant time-dependent behavior as well as plasticity is observed, which for metals typically occurs at elevated temperatures;
- consists of an elastic-plastic network that is in parallel with an elastic-viscous network (in contrast to the coupled creep and plasticity capabilities in which the plastic and the viscous networks are in series);
- is based on a Mises or Hill yield condition in the elastic-plastic network and any of the available creep models in Abaqus/Standard (except the hyperbolic creep law) in the elastic-viscous network;
- assumes a deviatoric inelastic response (hence, the pressure-dependent plasticity or creep models cannot be used to define the behavior of the two networks);
- is intended for modeling material response under fluctuating loads over a wide range of temperatures; and
- has been shown to provide good results for thermomechanical loading.

Material behavior

The material behavior is broken down into three parts: elastic, plastic, and viscous. Figure 23.2.11–1 shows a one-dimensional idealization of this material model, with the elastic-plastic and the elastic-viscous networks in parallel. The following subsections describe the elastic and the inelastic (plastic and viscous) behavior in detail.

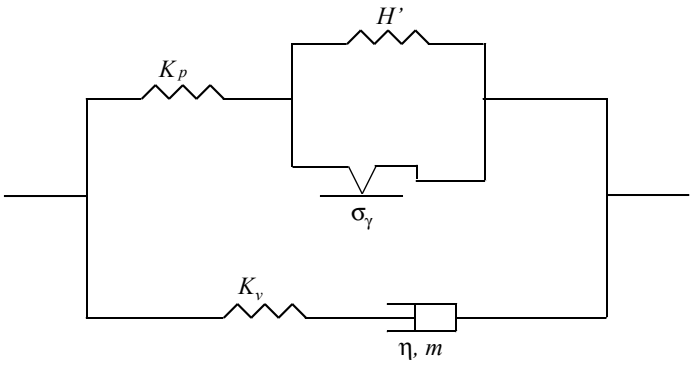


Figure 23.2.11–1 One-dimensional idealization of the two-layer viscoplasticity model.

Elastic behavior

The elastic part of the response for both networks is specified using a linear isotropic elasticity definition. Any one of the available elasticity models in Abaqus/Standard can be used to define the elastic behavior of the networks. Referring to the one-dimensional idealization (Figure 23.2.11–1), the ratio of the elastic modulus of the elastic-viscous network (K_V) to the total (instantaneous) modulus ($K_P + K_V$) is given by

$$f = \frac{K_V}{(K_P + K_V)}.$$

The user-specified ratio ***f***, given as part of the viscous behavior definition as discussed later, apportions the total moduli specified for the elastic behavior among the elastic-viscous and the elastic-plastic networks. As a result, if isotropic elastic properties are defined, the Poisson's ratios are the same in both networks. On the other hand, if anisotropic elasticity is defined, the same type of anisotropy holds for both networks. The properties specified for the elastic behavior are assumed to be the instantaneous properties ($K_P + K_V$).

Input File Usage: *ELASTIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**

Plastic behavior

A plasticity definition can be used to provide the static hardening data for the material model. All available metal plasticity models, including Hill's plasticity model to define anisotropic yield ("Anisotropic yield/creep," Section 23.2.6), can be used.

The elastic-plastic network does not take into account rate-dependent yield. Hence, any specification of strain rate dependence for the plasticity model is not allowed.

Input File Usage: Use the following options:

*PLASTIC
*POTENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Potential**

Viscous behavior

The viscous behavior of the material can be governed by any of the available creep laws in Abaqus/Standard ("Rate-dependent plasticity: creep and swelling," Section 23.2.4), except the hyperbolic creep law. When you define the viscous behavior, you specify the viscosity parameters and choose the specific type of viscous behavior. If you choose to input the creep law through user subroutine **CREEP**, only deviatoric creep should be defined—more specifically, volumetric swelling behavior should not be defined within user subroutine **CREEP**. In addition, you also specify the fraction, f, that defines the ratio of the elastic modulus of the elastic-viscous network to the total (instantaneous) modulus. Viscous stress ratios can be specified under the viscous behavior definition to define anisotropic viscosity (see "Anisotropic yield/creep," Section 23.2.6).

All material properties can be specified as functions of temperature and predefined field variables.

Input File Usage: Use the following options:

*VISCOUS, LAW=TIME *or* STRAIN *or* USER
*POTENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Viscous**: **Suboptions→Potential**

Thermal expansion

Thermal expansion can be modeled by providing the thermal expansion coefficient of the material ("Thermal expansion," Section 26.1.2). Anisotropic expansion can be defined in the usual manner. In the one-dimensional idealization the expansion element is assumed to be in series with the rest of the network.

Calibration of material parameters

The calibration procedure is best explained in the context of the one-dimensional idealization of the material model. In the following discussion the viscous behavior is assumed to be governed by the Norton-Hoff rate law, which is given by

$$\dot{\varepsilon}_V^v = A\sigma_V^n.$$

In the expression above the subscript V denotes quantities in the elastic-viscous network alone. This form of the rate law may be chosen, for example, by choosing a time-hardening power law for the viscous behavior and setting $m = 0$. For this basic case there are six material parameters that need to

be calibrated (Figure 23.2.11–1). These are the elastic properties of the two networks, K_P and K_V; the initial yield stress σ_y; the hardening $H^{'}$; and the Norton-Hoff rate parameters, $\boldsymbol{A}$ and $\boldsymbol{n}$.

The experiment that needs to be performed is uniaxial tension under different constant strain rates. A static (effectively zero strain rate) uniaxial tension test determines the long-term modulus, K_P; the initial yield stress, σ_y; and the hardening, $H^{'}$. The hardening is assumed to be linear for illustration purposes. The material model is not limited to linear hardening, and any general hardening behavior can be defined for the plasticity model. The instantaneous elastic modulus, $K = K_P + K_V$, can be measured by measuring the initial elastic response of the material under nonzero, relatively high, strain rates. Several such measurements at different applied strain rates can be compared until the instantaneous moduli does not change with a change in the applied strain rate. The difference between $\boldsymbol{K}$ and K_P determines K_V.

To calibrate the parameters $\boldsymbol{A}$ and $\boldsymbol{n}$, it is useful to recognize that the long-term (steady-state) behavior of the elastic-viscous network under a constant applied strain rate, $\dot{\varepsilon}_o$, is a constant stress of magnitude $\sigma_V = A^{-\frac{1}{n}} \dot{\varepsilon}_o^{\frac{1}{n}}$. Under the assumption that the hardening modulus is negligible compared to the elastic modulus ($K_P >> H^{'}$), the steady-state response of the overall material is given by

$$\sigma = A^{-\frac{1}{n}} \dot{\varepsilon}_o^{\frac{1}{n}} + \sigma_y + H^{'} \varepsilon,$$

where σ is the total stress for a given total strain ε. To determine whether steady state has been reached, one can plot the quantity $\bar{\sigma} = \sigma - \sigma_y - H^{'} \varepsilon$ as a function of ε and note when it becomes a constant. The constant value of $\bar{\sigma}$ is equal to $A^{-\frac{1}{n}} \dot{\varepsilon}_o^{\frac{1}{n}}$. By performing several tests at different values of the constant applied strain rate $\dot{\varepsilon}_o$, it is possible to determine the constants $\boldsymbol{A}$ and $\boldsymbol{n}$.

Material response in different analysis steps

The material is active during all stress/displacement procedure types. In a static analysis step where the long-term response is requested (see "Static stress analysis," Section 6.2.2), only the elastic-plastic network will be active; the elastic-viscous network will not contribute in any manner. In particular, the stress in the viscous network will be zero during a long-term static response. If the creep effects are removed in a coupled temperature-displacement procedure or a soils consolidation procedure, the response of the elastic-viscous network will be assumed to be elastic only. During a linear perturbation step, only the elastic response of the networks is considered.

Some stress/displacement procedure types (coupled temperature-displacement, soils consolidation, and quasi-static) allow user control of the time integration accuracy of the viscous constitutive equations through a user-specified error tolerance. In other procedure types where no such direct control is currently available (static, dynamic), you must choose appropriate time increments. These time increments must be small compared to the typical relaxation time of the material.

Elements

The two-layer viscoplastic model is not available for one-dimensional elements (beams and trusses). It can be used with any other element in Abaqus/Standard that includes mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus/Standard ("Abaqus/Standard output variable identifiers," Section 4.2.1), the following variables have special meaning for the two-layer viscoplastic material model:

EE	The elastic strain is defined as: $\boldsymbol{\varepsilon}^{el} = f\boldsymbol{\varepsilon}_V^{el} + (1-f)\boldsymbol{\varepsilon}_P^{el}$.
PE	Plastic strain, $\boldsymbol{\varepsilon}_P^{pl}$, in the elastic-plastic network.
VE	Viscous strain, $\boldsymbol{\varepsilon}_V^{v}$, in the elastic-viscous network.
PS	Stress, $\boldsymbol{\sigma}_P$, in the elastic-plastic network.
VS	Stress, $\boldsymbol{\sigma}_V$, in the elastic-viscous network.
PEEQ	The equivalent plastic strain, defined as $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}_P^{pl} : \dot{\boldsymbol{\varepsilon}}_P^{pl}}\, dt$.
VEEQ	The equivalent viscous strain, defined as $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}_V^{v} : \dot{\boldsymbol{\varepsilon}}_V^{v}}\, dt$.
SENER	The elastic strain energy density per unit volume, defined as $\frac{1}{2}\boldsymbol{\sigma}_P : \boldsymbol{\varepsilon}_P^{el} + \frac{1}{2}\boldsymbol{\sigma}_V : \boldsymbol{\varepsilon}_V^{el}$.
PENER	The plastic dissipated energy per unit volume, defined as $\int_0^t \boldsymbol{\sigma}_P : \dot{\boldsymbol{\varepsilon}}_P^{pl}\, dt$.
VENER	The viscous dissipated energy per unit volume, defined as $\int_0^t \boldsymbol{\sigma}_V : \dot{\boldsymbol{\varepsilon}}_V^{v}\, dt$.

The above definitions of the strain tensors imply that the total strain is related to the elastic, plastic, and viscous strains through the following relation:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + (1-f)\boldsymbol{\varepsilon}^{pl} + f\boldsymbol{\varepsilon}^{v},$$

where according to the definitions given above $\boldsymbol{\varepsilon}^{pl} = \boldsymbol{\varepsilon}_P^{pl}$ and $\boldsymbol{\varepsilon}^{v} = \boldsymbol{\varepsilon}_V^{v}$. The above definitions of the output variables apply to all procedure types. In particular, when the long-term response is requested for a static procedure, the elastic-viscous network does not carry any stress and the definition of the elastic strain reduces to $\boldsymbol{\varepsilon}^{el} = (1-f)\boldsymbol{\varepsilon}_P^{el}$, which implies that the total stress is related to the elastic strain through the instantaneous elastic moduli.

Additional reference

- Kichenin, J., "Comportement Thermomécanique du Polyéthylène—Application aux Structures Gazières," Thèse de Doctorat de l'Ecole Polytechnique, Spécialité: Mécanique et Matériaux, 1992.

23.2.12 ORNL – OAK RIDGE NATIONAL LABORATORY CONSTITUTIVE MODEL

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "Classical metal plasticity," Section 23.2.1
- "Rate-dependent plasticity: creep and swelling," Section 23.2.4
- *ORNL
- *PLASTIC
- *CREEP
- "Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual
- "Specifying cycled yield stress data for the ORNL model" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The Oak Ridge National Laboratory (ORNL) constitutive model:

- allows for use of the rules defined in the Nuclear Standard NEF 9–5T, "Guidelines and Procedures for Design of Class 1 Elevated Temperature Nuclear System Components," in plasticity and creep calculations;
- is intended for use in modeling types 304 and 316 stainless steel at relatively high temperatures;
- can be used only with the metal plasticity models (linear kinematic hardening only) and/or the strain hardening form of the metal creep law; and
- is described in detail in "ORNL constitutive theory," Section 4.3.8 of the Abaqus Theory Manual.

Usage with plasticity

The ORNL constitutive model in Abaqus/Standard is based on the March 1981 issue of the Nuclear Standard NEF 9–5T and on the October 1986 issue, which revises the constitutive model extensively. This model adds isotropic hardening of the plastic yield surface from a virgin material state to a fully cycled state. Initially the material is assumed to harden kinematically according to a bilinear representation of the virgin stress-strain curve. If a strain reversal takes place or if the creep strain reaches 0.2%, the yield surface expands isotropically to the user-defined tenth-cycle stress-strain curve. Further hardening occurs kinematically according to a bilinear representation of the tenth-cycle stress-strain curve.

You must specify the virgin yield stress and the hardening through a plasticity model definition and the elastic part of the response through a linear elasticity model definition. You specify the tenth-cycle yield stress and hardening values separately. The yield stress at each temperature should be defined by giving its value at zero plastic strain and at one additional nonzero plastic strain point, thus giving a constant hardening rate (linear work hardening).

Input File Usage: Use all of the following options in the same material data block:

*PLASTIC
*ORNL
*CYCLED PLASTIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**: **Suboptions→Ornl** and **Suboptions→Cycled Plastic**

Abaqus/Standard also allows you to invoke the optional kinematic shift (α) reset procedure that is described in Section 4.3.5 of the Nuclear Standard. If you do not specify the α reset procedure explicitly, it is not used.

Input File Usage: *ORNL, RESET

Abaqus/CAE Usage: Property module: material editor: **Suboptions→Ornl**: **Invoke reset procedure**

Usage with creep

The ORNL constitutive model assumes that creep uses the strain hardening formulation. It introduces auxiliary hardening rules when strain reversals occur. An algorithm providing details is presented in "ORNL constitutive theory," Section 4.3.8 of the Abaqus Theory Manual. It can be used only when the creep behavior is defined by a strain-hardening power law.

Input File Usage: Use both of the following options in the same material data block:

*CREEP, LAW=STRAIN
*ORNL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Creep**: **Law: Strain-Hardening**: **Suboptions→Ornl**

Translation of the yield surface during creep

The ORNL formulation can also cause the center of the yield surface to translate during creep for use in subsequent plastic increments; this behavior is defined through two optional user-defined parameters.

Specifying saturation rates for kinematic shift

You can specify A, the saturation rates for kinematic shift caused by creep strain as defined by Equation (15) of Section 4.3.3–3 of the Nuclear Standard. The default value is 0.3. Set A=0.0 to use the 1986 revision of the standard.

Input File Usage: *ORNL, A=A

Abaqus/CAE Usage: Property module: material editor: **Suboptions→Ornl**: **Saturation rates for kinematic shift:** A

Specifying the rate of kinematic shift

You can specify H, the rate of kinematic shift with respect to creep strain (Equation (7) of Section 4.3.2–1 of the Nuclear Standard). Set H=0.0 to use the 1986 revision of the standard. If you do not specify a value for H, it is determined according to Section 4.3.3–3 of the 1981 revision of the standard.

Input File Usage: *ORNL, H=H

Abaqus/CAE Usage: Property module: material editor: **Suboptions→Ornl**: **Rate of kinematic shift wrt creep strain:** H

Initial conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, initial equivalent plastic strain values can be provided to specify the yield stress corresponding to the work hardened state. See "Inelastic behavior," Section 23.1.1, for additional details. Initial values can also be provided for the backstress tensor, α, to include strain-induced anisotropy. See "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1, for more information. For more complicated cases initial conditions can be defined through user subroutine **HARDINI**.

Input File Usage: Use the following option to specify the initial equivalent plastic strain directly:

*INITIAL CONDITIONS, TYPE=HARDENING

Use the following option in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine **HARDINI**:

*INITIAL CONDITIONS, TYPE=HARDENING, USER

Abaqus/CAE Usage: Use the following options to specify the initial equivalent plastic strain directly:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use the following options in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine **HARDINI**:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

The ORNL constitutive model can be used with any elements in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus/Standard ("Abaqus/Standard output variable identifiers," Section 4.2.1), variables associated with creep ("Rate-dependent plasticity: creep and swelling," Section 23.2.4) and the kinematic hardening plasticity models ("Models for metals subjected to cyclic loading," Section 23.2.2) are available for the ORNL constitutive model.

23.2.13 DEFORMATION PLASTICITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *DEFORMATION PLASTICITY
- "Defining deformation plasticity" in "Defining other mechanical models," Section 12.9.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The deformation theory Ramberg-Osgood plasticity model:

- is primarily intended for use in developing fully plastic solutions for fracture mechanics applications in ductile metals; and
- cannot appear with any other mechanical response material models since it completely describes the mechanical response of the material.

One-dimensional model

In one dimension the model is

$$E\,\varepsilon = \sigma + \alpha \left(\frac{|\sigma|}{\sigma^0}\right)^{n-1} \sigma,$$

where

σ	is the stress;
ε	is the strain;
E	is Young's modulus (defined as the slope of the stress-strain curve at zero stress);
α	is the "yield" offset;
σ^0	is the yield stress, in the sense that, when $\sigma = \sigma^0$, $\varepsilon = (1+\alpha)\sigma^0/E$; and
n	is the hardening exponent for the "plastic" (nonlinear) term: $n > 1$.

The material behavior described by this model is nonlinear at all stress levels, but for commonly used values of the hardening exponent ($n \sim 5$ or more) the nonlinearity becomes significant only at stress magnitudes approaching or exceeding σ^0.

Generalization to multiaxial stress states

The one-dimensional model is generalized to multiaxial stress states using Hooke's law for the linear term and the Mises stress potential and associated flow law for the nonlinear term:

$$E\,\boldsymbol{\varepsilon} = (1+\nu)\,\mathbf{S} - (1-2\nu)p\,\mathbf{I} + \frac{3}{2}\alpha\left(\frac{q}{\sigma^0}\right)^{n-1}\mathbf{S},$$

where

$\boldsymbol{\varepsilon}$	is the strain tensor,
$\boldsymbol{\sigma}$	is the stress tensor,
$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$	is the equivalent hydrostatic stress,
$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$	is the Mises equivalent stress,
$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}$	is the stress deviator, and
ν	is the Poisson's ratio.

The linear part of the behavior can be compressible or incompressible, depending on the value of the Poisson's ratio, but the nonlinear part of the behavior is incompressible (because the flow is normal to the Mises stress potential). The model is described in detail in "Deformation plasticity," Section 4.3.9 of the Abaqus Theory Manual.

You specify the parameters E, ν, σ^0, n, and α directly. They can be defined as a tabular function of temperature.

Input File Usage: *DEFORMATION PLASTICITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Deformation Plasticity**

Typical applications

The deformation plasticity model is most commonly applied in static loading with small-displacement analysis, where the fully plastic solution must be developed in a part of the model. Generally, the load is ramped on until all points in the region being monitored satisfy the condition that the "plastic strain" dominates and, hence, exhibit fully plastic behavior, which is defined as

$$\frac{\alpha}{E}\left(\frac{q}{\sigma^0}\right)^{n-1} q > 10\frac{q}{E},$$

or

$$q > \left(\frac{10}{\alpha}\right)^{1/(n-1)}\sigma^0.$$

You can specify the name of a particular element set to be monitored in a static analysis step for fully plastic behavior. The step will end when the solutions at all constitutive calculation points in the element set are fully plastic, when the maximum number of increments specified for the step is reached, or when the time period specified for the static step is exceeded, whichever comes first.

Input File Usage: *STATIC, FULLY PLASTIC=*ElsetName*

Abaqus/CAE Usage: Step module: **Create Step**: **General**: **Static, General**: **Other**: **Stop when region *region* is fully plastic.**

Elements

Deformation plasticity can be used with any stress/displacement element in Abaqus/Standard. Since it will generally be used for cases when the deformation is dominated by plastic flow, the use of "hybrid" (mixed formulation) or reduced-integration elements is recommended with this material model.

23.3 Other plasticity models

- "Extended Drucker-Prager models," Section 23.3.1
- "Modified Drucker-Prager/Cap model," Section 23.3.2
- "Mohr-Coulomb plasticity," Section 23.3.3
- "Critical state (clay) plasticity model," Section 23.3.4
- "Crushable foam plasticity models," Section 23.3.5

23.3.1 EXTENDED DRUCKER-PRAGER MODELS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "Rate-dependent yield," Section 23.2.3
- "Rate-dependent plasticity: creep and swelling," Section 23.2.4
- Chapter 24, "Progressive Damage and Failure"
- *DRUCKER PRAGER
- *DRUCKER PRAGER HARDENING
- *RATE DEPENDENT
- *DRUCKER PRAGER CREEP
- *TRIAXIAL TEST DATA
- "Defining Drucker-Prager plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The extended Drucker-Prager models:

- are used to model frictional materials, which are typically granular-like soils and rock, and exhibit pressure-dependent yield (the material becomes stronger as the pressure increases);
- are used to model materials in which the compressive yield strength is greater than the tensile yield strength, such as those commonly found in composite and polymeric materials;
- allow a material to harden and/or soften isotropically;
- generally allow for volume change with inelastic behavior: the flow rule, defining the inelastic straining, allows simultaneous inelastic dilation (volume increase) and inelastic shearing;
- can include creep in Abaqus/Standard if the material exhibits long-term inelastic deformations;
- can be defined to be sensitive to the rate of straining, as is often the case in polymeric materials;
- can be used in conjunction with either the elastic material model ("Linear elastic behavior," Section 22.2.1) or, in Abaqus/Standard if creep is not defined, the porous elastic material model ("Elastic behavior of porous materials," Section 22.3.1);
- can be used in conjunction with an equation of state model ("Equation of state," Section 25.2.1) to describe the hydrodynamic response of the material in Abaqus/Explicit;
- can be used in conjunction with the models of progressive damage and failure ("Damage and failure for ductile metals: overview," Section 24.2.1) to specify different damage initiation criteria and

damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh; and

- are intended to simulate material response under essentially monotonic loading.

Yield criteria

The yield criteria for this class of models are based on the shape of the yield surface in the meridional plane. The yield surface can have a linear form, a hyperbolic form, or a general exponent form. These surfaces are illustrated in Figure 23.3.1–1. The stress invariants and other terms in each of the three related yield criteria are defined later in this section.

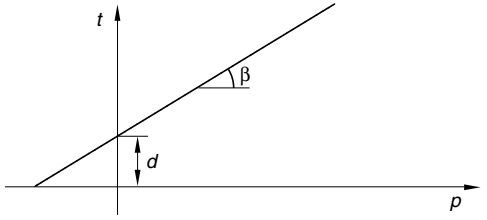
The linear model (Figure 23.3.1–1a) provides for a possibly noncircular yield surface in the deviatoric plane (Π-plane) to match different yield values in triaxial tension and compression, associated inelastic flow in the deviatoric plane, and separate dilation and friction angles. Input data parameters define the shape of the yield and flow surfaces in the meridional and deviatoric planes as well as other characteristics of inelastic behavior such that a range of simple theories is provided—the original Drucker-Prager model is available within this model. However, this model cannot provide a close match to Mohr-Coulomb behavior, as described later in this section.

The hyperbolic and general exponent models use a von Mises (circular) section in the deviatoric stress plane. In the meridional plane a hyperbolic flow potential is used for both models, which, in general, means nonassociated flow.

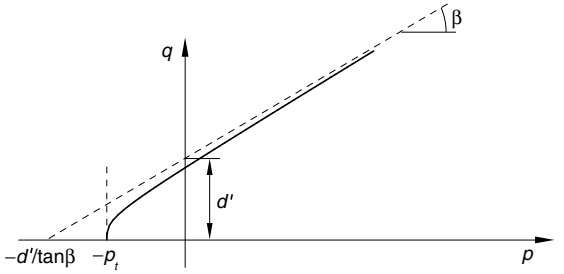
The choice of model to be used depends largely on the analysis type, the kind of material, the experimental data available for calibration of the model parameters, and the range of pressure stress values that the material is likely to experience. It is common to have either triaxial test data at different levels of confining pressure or test data that are already calibrated in terms of a cohesion and a friction angle and, sometimes, a triaxial tensile strength value. If triaxial test data are available, the material parameters must be calibrated first. The accuracy with which the linear model can match these test data is limited by the fact that it assumes linear dependence of deviatoric stress on pressure stress. Although the hyperbolic model makes a similar assumption at high confining pressures, it provides a nonlinear relationship between deviatoric and pressure stress at low confining pressures, which may provide a better match of the triaxial experimental data. The hyperbolic model is useful for brittle materials for which both triaxial compression and triaxial tension data are available, which is a common situation for materials such as rocks. The most general of the three yield criteria is the exponent form. This criterion provides the most flexibility in matching triaxial test data. Abaqus determines the material parameters required for this model directly from the triaxial test data. A least-squares fit that minimizes the relative error in stress is used for this purpose.

For cases where the experimental data are already calibrated in terms of a cohesion and a friction angle, the linear model can be used. If these parameters are provided for a Mohr-Coulomb model, it is necessary to convert them to Drucker-Prager parameters. The linear model is intended primarily for applications where the stresses are for the most part compressive. If tensile stresses are significant, hydrostatic tension data should be available (along with the cohesion and friction angle) and the hyperbolic model should be used.

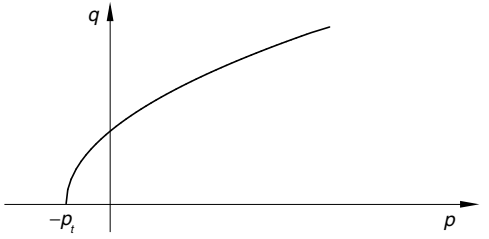
Calibration of these models is discussed later in this section.



a) Linear Drucker-Prager: $F = t - p \tan \beta - d = 0$



b) Hyperbolic: $F = \sqrt{(d'|_0 - p_t|_0 \tan \beta)^2 + q^2} - p \tan \beta - d' = 0$



c) Exponent form: $F = aq^b - p - p_t = 0$

Figure 23.3.1–1 Yield surfaces in the meridional plane.

Hardening and rate dependence

For granular materials these models are often used as a failure surface, in the sense that the material can exhibit unlimited flow when the stress reaches yield. This behavior is called perfect plasticity. The models are also provided with isotropic hardening. In this case plastic flow causes the yield surface to change size uniformly with respect to all stress directions. This hardening model is useful for cases involving gross plastic straining or in which the straining at each point is essentially in the same direction

in strain space throughout the analysis. Although the model is referred to as an isotropic "hardening" model, strain softening, or hardening followed by softening, can be defined.

As strain rates increase, many materials show an increase in their yield strength. This effect becomes important in many polymers when the strain rates range between 0.1 and 1 per second; it can be very important for strain rates ranging between 10 and 100 per second, which are characteristic of high-energy dynamic events or manufacturing processes. The effect is generally not as important in granular materials. The evolution of the yield surface with plastic deformation is described in terms of the equivalent stress $\bar{\sigma}$, which can be chosen as either the uniaxial compression yield stress, the uniaxial tension yield stress, or the shear (cohesion) yield stress:

$$\begin{aligned}\bar{\sigma} &= \sigma_c(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i) \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c;\\ &= \sigma_t(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i) \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t; \text{ or}\\ &= d(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i) \quad \text{if hardening is defined by the cohesion, } d,\end{aligned}$$

where

$\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate, defined for the linear Drucker-Prager model as

$\dot{\bar{\varepsilon}}^{pl}$ $=|\dot{\varepsilon}^{pl}_{11}|$ if hardening is defined in uniaxial compression;
$=\dot{\varepsilon}^{pl}_{11}$ if hardening is defined in uniaxial tension;
$=\dot{\gamma}^{pl}/\sqrt{3}$ if hardening is defined in pure shear,

and defined for the hyperbolic and exponential Drucker-Prager models as

$$\dot{\bar{\varepsilon}}^{pl} = \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}}{\bar{\sigma}};$$

$\bar{\varepsilon}^{pl} = \int_0^t \dot{\bar{\varepsilon}}^{pl} dt$ is the equivalent plastic strain;

θ is temperature; and

$f_i, i = 1, 2, \ldots$ are other predefined field variables.

The functional dependence $\bar{\sigma}(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i)$ includes hardening as well as rate-dependent effects. The material data can be input either directly in a tabular format or by correlating it to static relations based on yield stress ratios.

Rate dependence as described here is most suitable for moderate- to high-speed events in Abaqus/Standard. Time-dependent inelastic deformation at low deformation rates can be better represented by creep models. Such inelastic deformation, which can coexist with rate-independent plastic deformation, is described later in this section. However, the existence of creep in an Abaqus/Standard material definition precludes the use of rate dependence as described here.

When using the Drucker-Prager material model, Abaqus allows you to prescribe initial hardening by defining initial equivalent plastic strain values, as discussed below along with other details regarding the use of initial conditions.

Direct tabular data

Test data are entered as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates; one table per strain rate. Compression data are more commonly available for geological materials, whereas tension data are usually available for polymeric materials. The guidelines on how to enter these data are provided in "Rate-dependent yield," Section 23.2.3.

Input File Usage: *DRUCKER PRAGER HARDENING, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Hardening**: toggle on **Use strain-rate-dependent data**

Yield stress ratios

Alternatively, the strain rate behavior can be assumed to be separable, so that the stress-strain dependence is similar at all strain rates:

$$\overline{\sigma} = \sigma^0(\bar{\varepsilon}^{pl}, \theta, f_i) R(\dot{\bar{\varepsilon}}^{pl}, \theta, f_i),$$

where $\sigma^0(\bar{\varepsilon}^{pl}, \theta, f_i)$ is the static stress-strain behavior and $R(\dot{\bar{\varepsilon}}^{pl}, \theta, f_i)$ is the ratio of the yield stress at nonzero strain rate to the static yield stress (so that $R(0, \theta, f_i) = 1.0$).

Two methods are offered to define $\boldsymbol{R}$ in Abaqus: specifying an overstress power law or defining the variable $\boldsymbol{R}$ directly as a tabular function of $\dot{\bar{\varepsilon}}^{pl}$.

Overstress power law

The Cowper-Symonds overstress power law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D\,(R-1)^n \quad \text{for} \quad \bar{\sigma} \geq \sigma^0,$$

where $D(\theta, f_i)$ and $n(\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: Use both of the following options:

*DRUCKER PRAGER HARDENING
*RATE DEPENDENT, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Hardening**; **Suboptions→Rate Dependent**: **Hardening: Power Law**

Tabular function

When R is entered directly, it is entered as a tabular function of the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and predefined field variables, f_i.

Input File Usage: Use both of the following options:

*DRUCKER PRAGER HARDENING
*RATE DEPENDENT, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Hardening**; **Suboptions→Rate Dependent**: **Hardening: Yield Ratio**

Johnson-Cook rate dependence

Johnson-Cook rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \exp\left[\frac{1}{C}(R-1)\right] \quad \text{for} \quad \bar{\sigma} \geq \sigma^0,$$

where $\dot{\varepsilon}_0$ and C are material constants that do not depend on temperature and are assumed not to depend on predefined field variables.

Input File Usage: *DRUCKER PRAGER HARDENING
*RATE DEPENDENT, TYPE=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Hardening**; **Suboptions→Rate Dependent**: **Hardening: Johnson-Cook**

Stress invariants

The yield stress surface makes use of two invariants, defined as the equivalent pressure stress,

$$p = -\frac{1}{3}\text{trace}\,(\boldsymbol{\sigma}),$$

and the Mises equivalent stress,

$$q = \sqrt{\frac{3}{2}(\mathbf{S} : \mathbf{S})},$$

where $\mathbf{S}$ is the stress deviator, defined as

$$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}.$$

In addition, the linear model also uses the third invariant of deviatoric stress,

$$r = (\frac{9}{2}\mathbf{S}\cdot\mathbf{S}:\mathbf{S})^{\frac{1}{3}}.$$

Linear Drucker-Prager model

The linear model is written in terms of all three stress invariants. It provides for a possibly noncircular yield surface in the deviatoric plane to match different yield values in triaxial tension and compression, associated inelastic flow in the deviatoric plane, and separate dilation and friction angles.

Yield criterion

The linear Drucker-Prager criterion (see Figure 23.3.1–1a) is written as

$$F = t - p\tan\beta - d = 0,$$

where

$$t = \frac{1}{2}q\left[1 + \frac{1}{K} - \left(1 - \frac{1}{K}\right)\left(\frac{r}{q}\right)^3\right].$$

$\beta(\theta, f_i)$ is the slope of the linear yield surface in the ***p*–*t*** stress plane and is commonly referred to as the friction angle of the material;

d is the cohesion of the material; and

$K(\theta, f_i)$ is the ratio of the yield stress in triaxial tension to the yield stress in triaxial compression and, thus, controls the dependence of the yield surface on the value of the intermediate principal stress (see Figure 23.3.1–2).

In the case of hardening defined in uniaxial compression, the linear yield criterion precludes friction angles $\beta > 71.5°$ ($\tan\beta > 3$), which is unlikely to be a limitation for real materials.

When $K = 1,\ t = q$, which implies that the yield surface is the von Mises circle in the deviatoric principal stress plane (the Π-plane), in which case the yield stresses in triaxial tension and compression are the same. To ensure that the yield surface remains convex requires $0.778 \le K \le 1.0$.

The cohesion, ***d***, of the material is related to the input data as

$$\begin{aligned} d &= (1 - \frac{1}{3}\tan\beta)\sigma_c \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c; \\ &= (\frac{1}{K} + \frac{1}{3}\tan\beta)\sigma_t \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t; \\ &= d \quad \text{if hardening is defined by the cohesion, } d = \frac{\sqrt{3}}{2}\,\tau(1 + \frac{1}{K}). \end{aligned}$$

Plastic flow

G is the flow potential, chosen in this model as

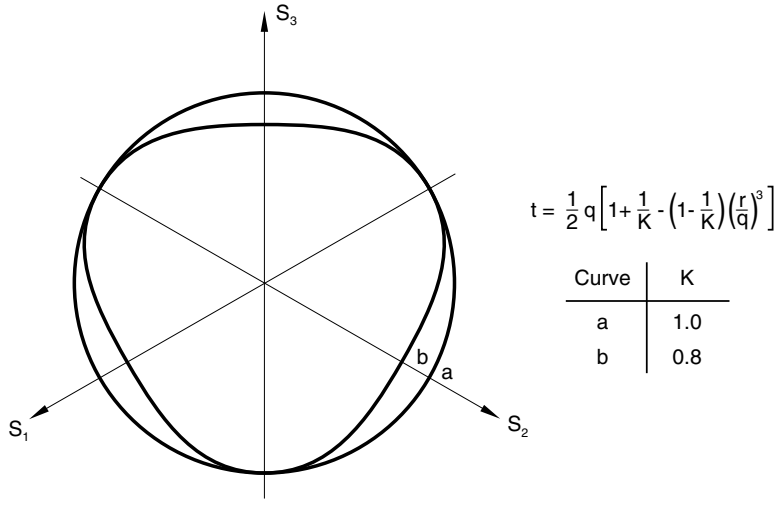


Figure 23.3.1–2 Typical yield/flow surfaces of the linear model in the deviatoric plane.

$$G = t - p \tan \psi,$$

where $\psi(\theta, f_i)$ is the dilation angle in the p–t plane. A geometric interpretation of ψ is shown in the p–t diagram of Figure 23.3.1–3. In the case of hardening defined in uniaxial compression, this flow rule definition precludes dilation angles $\psi > 71.5°$ ($\tan \psi > 3$). This restriction is not seen as a limitation since it is unlikely this will be the case for real materials.

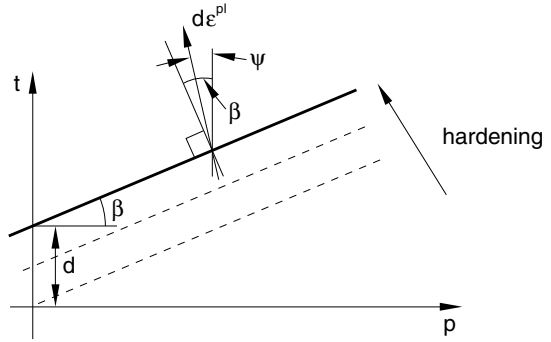


Figure 23.3.1–3 Linear Drucker-Prager model: yield surface and flow direction in the p–t plane.

For granular materials the linear model is normally used with nonassociated flow in the p–t plane, in the sense that the flow is assumed to be normal to the yield surface in the Π-plane but at an angle ψ

to the t-axis in the p–t plane, where usually $\psi < \beta$, as illustrated in Figure 23.3.1–3. Associated flow results from setting $\psi = \beta$. The original Drucker-Prager model is available by setting $\psi = \beta$ and $K = 1$. Nonassociated flow is also generally assumed when the model is used for polymeric materials. If $\psi = 0$, the inelastic deformation is incompressible; if $\psi \geq 0$, the material dilates. Hence, ψ is referred to as the dilation angle.

The relationship between the flow potential and the incremental plastic strain for the linear model is discussed in detail in "Models for granular or polymer behavior," Section 4.4.2 of the Abaqus Theory Manual.

Input File Usage: *DRUCKER PRAGER, SHEAR CRITERION=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Shear criterion: Linear**

Nonassociated flow

Nonassociated flow implies that the material stiffness matrix is not symmetric; therefore, the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see "Defining an analysis," Section 6.1.2). If the difference between β and ψ is not large and the region of the model in which inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence and the unsymmetric matrix scheme may not be needed.

Hyperbolic and general exponent models

The hyperbolic and general exponent models available are written in terms of the first two stress invariants only.

Hyperbolic yield criterion

The hyperbolic yield criterion is a continuous combination of the maximum tensile stress condition of Rankine (tensile cutoff) and the linear Drucker-Prager condition at high confining stress. It is written as

$$F = \sqrt{l_0^2 + q^2} - p \tan\beta - d' = 0,$$

where $l_0 = d'|_0 - p_t|_0 \tan\beta$ and

$p_t\|_0$	is the initial hydrostatic tension strength of the material;
$d'(\bar{\sigma})$	is the hardening parameter;
$d'\|_0$	is the initial value of d'; and
$\beta(\theta, f_i)$	is the friction angle measured at high confining pressure, as shown in Figure 23.3.1–1(b).

The hardening parameter, $d'(\bar{\sigma})$, can be obtained from test data as follows:

$$d' = \sqrt{l_0^2 + {\sigma_c}^2} - \frac{\sigma_c}{3}\tan\beta \text{ if hardening is defined by the uniaxial compression yield stress, } \sigma_c;$$
$$= \sqrt{l_0^2 + {\sigma_t}^2} + \frac{\sigma_t}{3}\tan\beta \text{ if hardening is defined by the uniaxial tension yield stress, } \sigma_t;$$
$$= \sqrt{l_0^2 + d^2} \quad \text{if hardening is defined by the cohesion, } d.$$

The isotropic hardening assumed in this model treats β as constant with respect to stress as depicted in Figure 23.3.1–4.

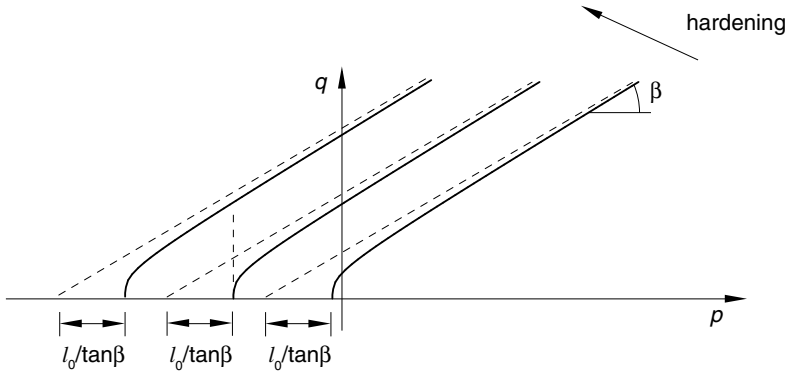


Figure 23.3.1–4 Hyperbolic model: yield surface and hardening in the p–q plane.

Input File Usage: *DRUCKER PRAGER, SHEAR CRITERION=HYPERBOLIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Shear criterion: Hyperbolic**

General exponent yield criterion

The general exponent form provides the most general yield criterion available in this class of models. The yield function is written as

$$F = aq^b - p - p_t = 0,$$

where

$a(\theta, f_i)$ and $b(\theta, f_i)$ are material parameters that are independent of plastic deformation; and

$p_t(\bar{\sigma})$ is the hardening parameter that represents the hydrostatic tension strength of the material as shown in Figure 23.3.1–1(c).

$p_t(\bar{\sigma})$ is related to the input test data as

$$p_t = a{\sigma_c}^b - \frac{\sigma_c}{3} \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c;$$
$$= a{\sigma_t}^b + \frac{\sigma_t}{3} \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t;$$
$$= ad^b \quad \text{if hardening is defined by the cohesion, } d.$$

The isotropic hardening assumed in this model treats a and b as constant with respect to stress, as depicted in Figure 23.3.1–5.

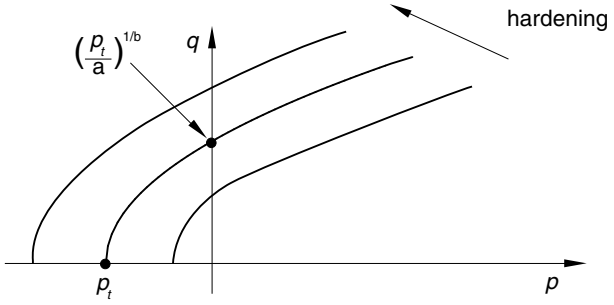


Figure 23.3.1–5 General exponent model: yield surface and hardening in the p–q plane.

The material parameters a and b can be given directly. Alternatively, if triaxial test data at different levels of confining pressure are available, Abaqus will determine the material parameters from the triaxial test data, as discussed below.

Input File Usage: *DRUCKER PRAGER, SHEAR CRITERION=EXPONENT FORM

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Shear criterion: Exponent Form**

Plastic flow

G is the flow potential, chosen in these models as a hyperbolic function:

$$G = \sqrt{(\epsilon\bar{\sigma}|_0 \tan\psi)^2 + q^2} - p\tan\psi,$$

where

$\psi(\theta, f_i)$ is the dilation angle measured in the p–q plane at high confining pressure;

$\bar{\sigma}|_0 = \bar{\sigma}|_{\bar{\varepsilon}^{pl}=0, \dot{\bar{\varepsilon}}^{pl}=0}$ is the initial yield stress, taken from the user-specified Drucker-Prager hardening data; and

ϵ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as the eccentricity tends to zero).

Suitable default values are provided for ϵ, as described below. The value of ϵ will depend on the yield stress used.

This flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches the linear Drucker-Prager flow potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 23.3.1–6. The flow potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

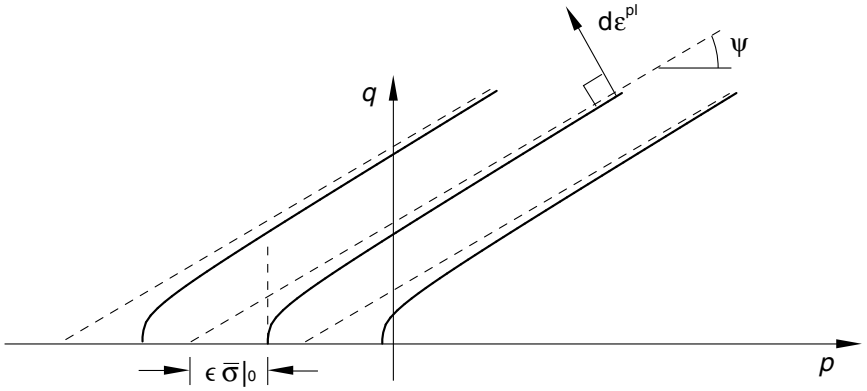


Figure 23.3.1–6 Family of hyperbolic flow potentials in the p–q plane.

For the hyperbolic model flow is nonassociated in the p–q plane if the dilation angle, ψ, and the material friction angle, β, are different. The hyperbolic model provides associated flow in the p–q plane only when $\beta = \psi$ and $d'|_0/\tan\beta - p_t|_0 = \epsilon\bar{\sigma}|_0$. A default value of $\epsilon = (d'|_0 - p_t|_0 \tan\beta)/(\bar{\sigma}|_0 \tan\beta)$ is assumed if the flow potential is used with the hyperbolic model, so that associated flow is recovered when $\psi = \beta$.

For the general exponent model flow is always nonassociated in the p–q plane. The default flow potential eccentricity is $\epsilon = 0.1$, which implies that the material has almost the same dilation angle over a wide range of confining pressure stress values. Increasing the value of ϵ provides more curvature to the flow potential, implying that the dilation angle increases more rapidly as the confining pressure decreases. Values of ϵ that are significantly less than the default value may lead to convergence problems if the material is subjected to low confining pressures because of the very tight curvature of the flow potential locally where it intersects the p-axis.

The relationship between the flow potential and the incremental plastic strain for the hyperbolic and general exponent models is discussed in detail in "Models for granular or polymer behavior," Section 4.4.2 of the Abaqus Theory Manual.

Nonassociated flow

Nonassociated flow implies that the material stiffness matrix is not symmetric; therefore, the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see "Defining an analysis," Section 6.1.2). If the difference between β and ψ in the hyperbolic model is not large and if the region of the model in which inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence. In such cases the unsymmetric matrix scheme may not be needed.

Progressive damage and failure

In Abaqus/Explicit the extended Drucker-Prager models can be used in conjunction with the models of progressive damage and failure discussed in "Damage and failure for ductile metals: overview," Section 24.2.1. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), and Müschenborn-Sonne forming limit diagram (MSFLD) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations.

Input File Usage: Use the following options:

*DAMAGE INITIATION
*DAMAGE EVOLUTION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*damage initiation type***: specify the damage initiation criterion: **Suboptions→Damage Evolution**: specify the damage evolution parameters

Matching experimental triaxial test data

Data for geological materials are most commonly available from triaxial testing. In such a test the specimen is confined by a pressure stress that is held constant during the test. The loading is an additional tension or compression stress applied in one direction. Typical results include stress-strain curves at different levels of confinement, as shown in Figure 23.3.1–7. To calibrate the yield parameters for this class of models, you need to decide which point on each stress-strain curve will be used for calibration. For example, if you wish to calibrate the initial yield surface, the point in each stress-strain curve corresponding to initial deviation from elastic behavior should be used. Alternatively, if you wish to calibrate the ultimate yield surface, the point in each stress-strain curve corresponding to the peak stress should be used.

One stress data point from each stress-strain curve at a different level of confinement is plotted in the meridional stress plane (p–t plane if the linear model is to be used, or p–q plane if the hyperbolic or general exponent model will be used). This technique calibrates the shape and position of the yield surface, as shown in Figure 23.3.1–8, and is adequate to define a model if it is to be used as a failure surface (perfect

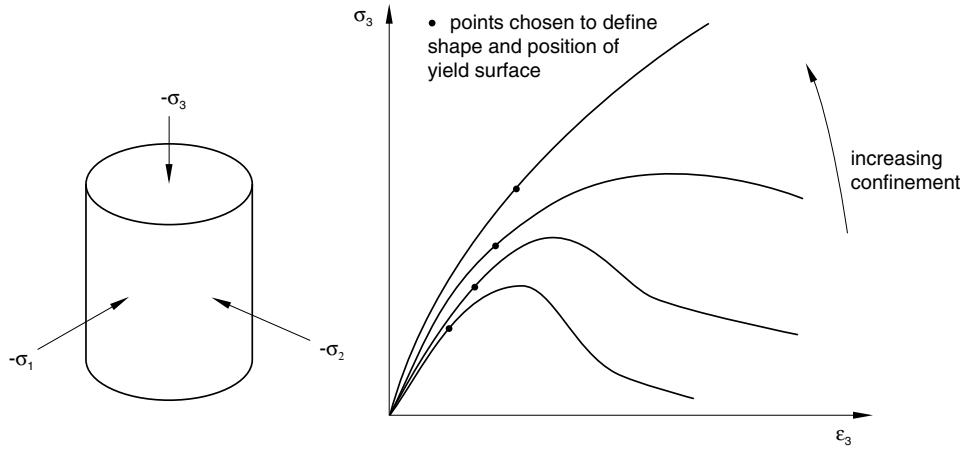


Figure 23.3.1–7 Triaxial tests with stress-strain curves for typical geological materials at different levels of confinement.

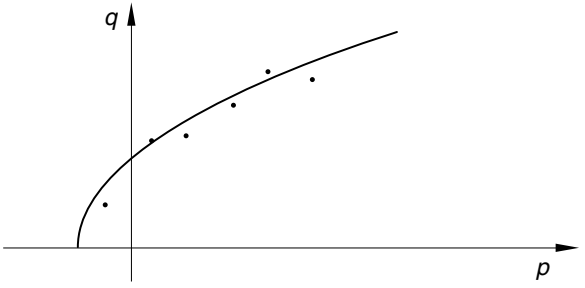


Figure 23.3.1–8 Yield surface in meridional plane.

plasticity). The models are also available with isotropic hardening, in which case hardening data are required to complete the calibration. In an isotropic hardening model plastic flow causes the yield surface to change size uniformly; in other words, only one of the stress-strain curves depicted in Figure 23.3.1–7 can be used to represent hardening. The curve that represents hardening most accurately over the range of loading conditions anticipated should be selected (usually the curve for the average anticipated value of pressure stress).

As stated earlier, two types of triaxial test data are commonly available for geological materials. In a triaxial compression test the specimen is confined by pressure and an additional compression stress is superposed in one direction. Thus, the principal stresses are all negative, with $0 \geq \sigma_1 = \sigma_2 \geq \sigma_3$ (Figure 23.3.1–9a). In the preceding inequality σ_1, σ_2, and σ_3 are the maximum, intermediate, and minimum principal stresses, respectively.

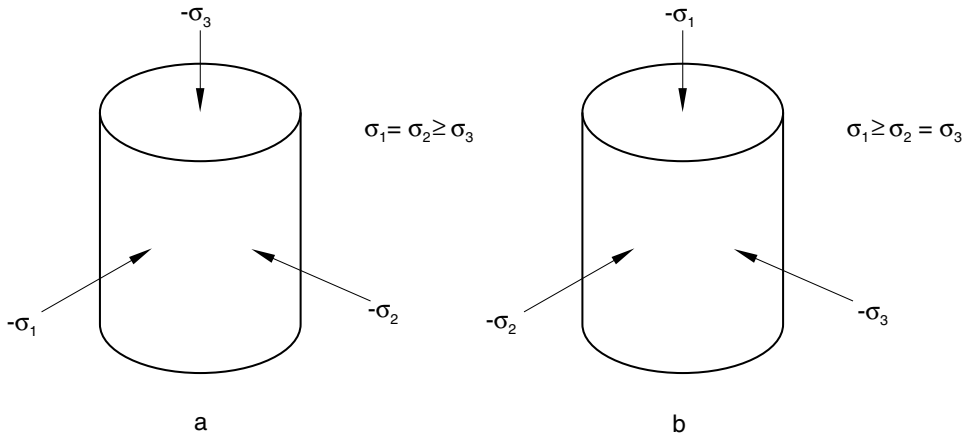


Figure 23.3.1–9 a) Triaxial compression and b) tension.

The values of the stress invariants are

$$p = -\frac{1}{3}(2\sigma_1 + \sigma_3),$$

$$q = \sigma_1 - \sigma_3,$$

and

$$r^3 = -(\sigma_1 - \sigma_3)^3,$$

so that

$$t = q = \sigma_1 - \sigma_3.$$

The triaxial compression results can, thus, be plotted in the meridional plane shown in Figure 23.3.1–8.

Linear Drucker-Prager model

Fitting the best straight line through the triaxial compression results provides β and $\boldsymbol{d}$ for the linear Drucker-Prager model.

Triaxial tension data are also needed to define $\boldsymbol{K}$ in the linear Drucker-Prager model. Under triaxial tension the specimen is again confined by pressure, after which the pressure in one direction is reduced. In this case the principal stresses are $0 \geq \sigma_1 \geq \sigma_2 = \sigma_3$ (Figure 23.3.1–9b).

The stress invariants are now

$$p = -\frac{1}{3}(\sigma_1 + 2\sigma_3),$$

$$q = \sigma_1 - \sigma_3,$$

and

$$r^3 = (\sigma_1 - \sigma_3)^3,$$

so that

$$t = \frac{q}{K} = \frac{1}{K}(\sigma_1 - \sigma_3).$$

Thus, K can be found by plotting these test results as q versus p and again fitting the best straight line. The triaxial compression and tension lines must intercept the p-axis at the same point, and the ratio of values of q for triaxial tension and compression at the same value of p then gives K (Figure 23.3.1–10).

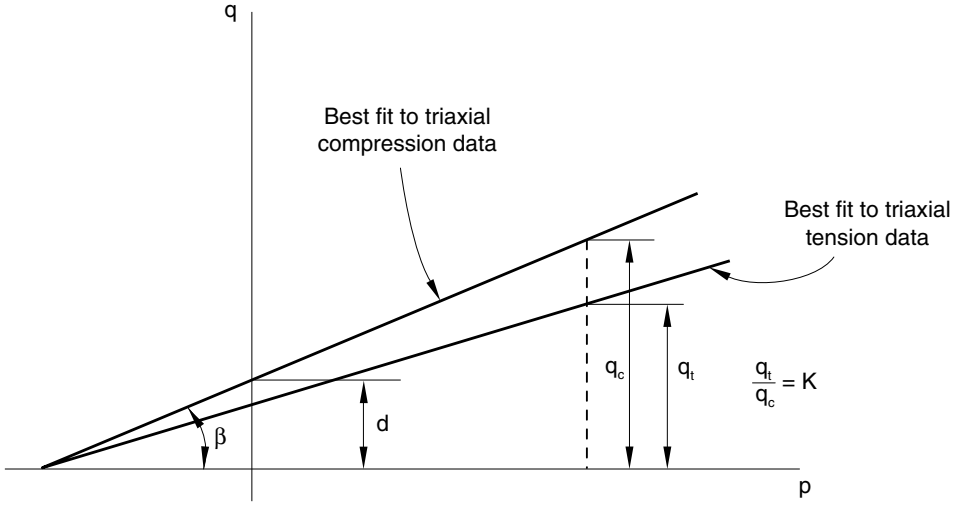


Figure 23.3.1–10 Linear model: fitting triaxial compression and tension data.

Hyperbolic model

Fitting the best straight line through the triaxial compression results at high confining pressures provides β and d' for the hyperbolic model. This fit is performed in the same manner as that used to obtain β and d for the linear Drucker-Prager model. In addition, hydrostatic tension data are required to complete the calibration of the hyperbolic model so that the initial hydrostatic tension strength, $p_t|_0$, can be defined.

General exponent model

Given triaxial data in the meridional plane, Abaqus provides a capability to determine the material parameters a, b, and p_t required for the exponent model. The parameters are determined on the basis of a "best fit" of the triaxial test data at different levels of confining stress. A least-squares fit which minimizes the relative error in stress is used to obtain the "best fit" values for a, b, and p_t. The capability allows all three parameters to be calibrated or, if some of the parameters are known, only the remaining parameters to be calibrated. This ability is useful if only a few data points are available, in which case you may wish to fit the best straight line ($b = 1$) through the data points (effectively reducing the model to a linear Drucker-Prager model). Partial calibration can also be useful in a case when triaxial test data at low confinement are unreliable or unavailable, as is often the case for cohesionless materials. In this case a better fit may be obtained if the value of p_t is specified and only a and b are calibrated.

The data must be provided in terms of the principal stresses $\sigma_1 (= \sigma_2)$ and σ_3, where σ_1 is the confining stress and σ_3 is the stress in the loading direction. The Abaqus sign convention must be followed such that tensile stresses are positive and compressive stresses are negative. One pair of stresses must be entered from each triaxial test. As many data points as desired can be entered from triaxial tests at different levels of confining stress.

If the exponent model is used as a failure surface (perfect plasticity), the Drucker-Prager hardening behavior does not have to be specified. The hydrostatic tension strength, p_t, obtained from the calibration will then be used as the failure stress. However, if the Drucker-Prager hardening behavior is specified together with the triaxial test data, the value of p_t obtained from the calibration will be ignored. In this case Abaqus will interpolate p_t directly from the hardening data.

Input File Usage: Use both of the following options:

```
*DRUCKER PRAGER, SHEAR CRITERION=EXPONENT FORM,
TEST DATA
*TRIAXIAL TEST DATA
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Shear criterion: Exponent Form**, toggle on **Use Suboption Triaxial Test Data**, and select **Suboptions→Triaxial Test Data**

Matching Mohr-Coulomb parameters to the Drucker-Prager model

Sometimes experimental data are not directly available. Instead, you are provided with the friction angle and cohesion values for the Mohr-Coulomb model. In that case the simplest way to proceed is to use the Mohr-Coulomb model (see "Mohr-Coulomb plasticity," Section 23.3.3). In some situations it may be necessary to use the Drucker-Prager model instead of the Mohr-Coulomb model (such as when rate effects need to be considered), in which case we need to calculate values for the parameters of a Drucker-Prager model to provide a reasonable match to the Mohr-Coulomb parameters.

The Mohr-Coulomb failure model is based on plotting Mohr's circle for states of stress at failure in the plane of the maximum and minimum principal stresses. The failure line is the best straight line that touches these Mohr's circles (Figure 23.3.1–11).

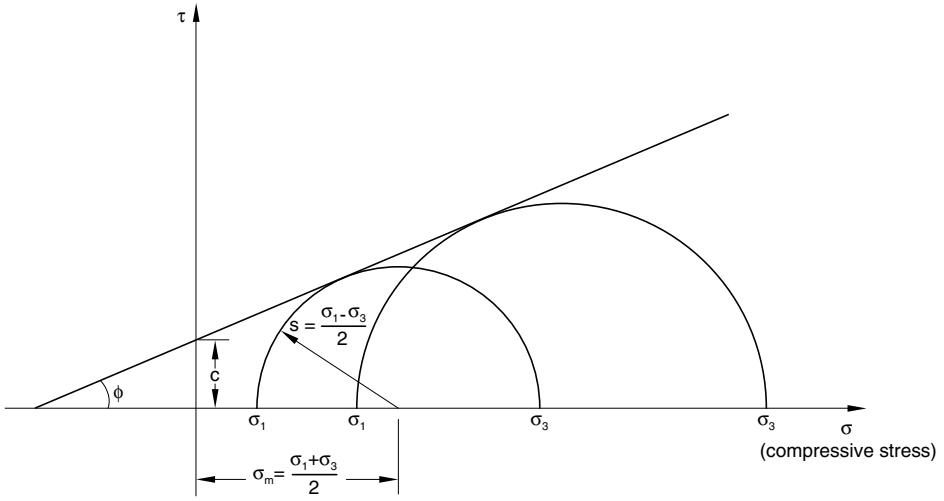


Figure 23.3.1–11 Mohr-Coulomb failure model.

Therefore, the Mohr-Coulomb model is defined by

$$\tau = c - \sigma \tan \phi,$$

where σ is negative in compression. From Mohr's circle,

$$\tau = s \cos \phi,$$
$$\sigma = \sigma_m + s \sin \phi.$$

Substituting for τ and σ, multiplying both sides by $\cos \phi$, and reducing, the Mohr-Coulomb model can be written as

$$s + \sigma_m \sin \phi - c \cos \phi = 0,$$

where

$$s = \frac{1}{2}(\sigma_1 - \sigma_3)$$

is half of the difference between the maximum principal stress, σ_1, and the minimum principal stress, σ_3 (and is, therefore, the maximum shear stress),

$$\sigma_m = \frac{1}{2}(\sigma_1 + \sigma_3)$$

is the average of the maximum and minimum principal stresses, and ϕ is the friction angle. Thus, the model assumes a linear relationship between deviatoric and pressure stress and, so, can be matched by the linear or hyperbolic Drucker-Prager models provided in Abaqus.

The Mohr-Coulomb model assumes that failure is independent of the value of the intermediate principal stress, but the Drucker-Prager model does not. The failure of typical geotechnical materials generally includes some small dependence on the intermediate principal stress, but the Mohr-Coulomb model is generally considered to be sufficiently accurate for most applications. This model has vertices in the deviatoric plane (see Figure 23.3.1–12).

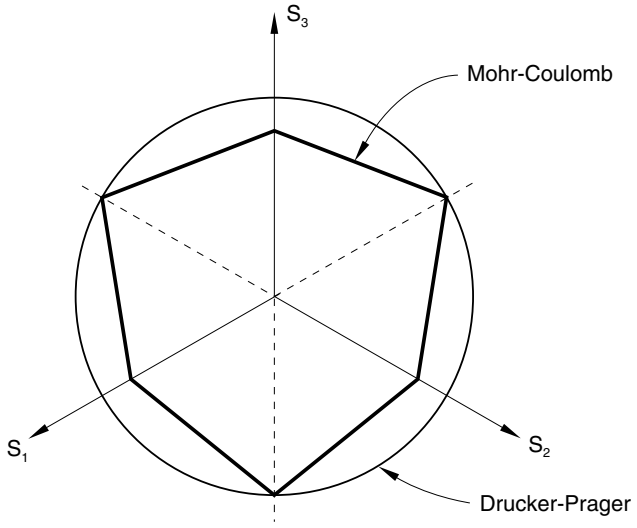


Figure 23.3.1–12 Mohr-Coulomb model in the deviatoric plane.

The implication is that, whenever the stress state has two equal principal stress values, the flow direction can change significantly with little or no change in stress. None of the models currently available in Abaqus can provide such behavior; even in the Mohr-Coulomb model the flow potential is smooth. This limitation is generally not a key concern in many design calculations involving Coulomb-like materials, but it can limit the accuracy of the calculations, especially in cases where flow localization is important.

Matching plane strain response

Plane strain problems are often encountered in geotechnical analysis; for example, long tunnels, footings, and embankments. Therefore, the constitutive model parameters are often matched to provide the same flow and failure response in plane strain.

The matching procedure described below is carried out in terms of the linear Drucker-Prager model but is also applicable to the hyperbolic model at high levels of confining stress.

The linear Drucker-Prager flow potential defines the plastic strain increment as

$$d\boldsymbol{\varepsilon}^{pl} = d\bar{\varepsilon}^{pl} \frac{1}{(1 - \frac{1}{3}\tan\psi)} \frac{\partial}{\partial \boldsymbol{\sigma}}(t - p\tan\psi),$$

where $d\bar{\varepsilon}^{pl}$ is the equivalent plastic strain increment. Since we wish to match the behavior in only one plane, we can take $K = 1$, which implies that $t = q$. Thus,

$$d\boldsymbol{\varepsilon}^{pl} = d\bar{\varepsilon}^{pl} \frac{1}{(1 - \frac{1}{3}\tan\psi)} \left(\frac{\partial q}{\partial \boldsymbol{\sigma}} - \tan\psi \frac{\partial p}{\partial \boldsymbol{\sigma}} \right).$$

Writing this expression in terms of principal stresses provides

$$d{\varepsilon_1}^{pl} = d\bar{\varepsilon}^{pl} \frac{1}{(1 - \frac{1}{3}\tan\psi)} \left(\frac{1}{2q}(2\sigma_1 - \sigma_2 - \sigma_3) + \frac{1}{3}\tan\psi \right),$$

with similar expressions for $d\varepsilon_2^{pl}$ and $d\varepsilon_3^{pl}$. Assume plane strain in the 1-direction. At limit load we must have $d{\varepsilon_1}^{pl} = 0$, which provides the constraint

$$\sigma_1 = \frac{1}{2}(\sigma_2 + \sigma_3) - \frac{1}{3}\tan\psi\, q.$$

Using this constraint we can rewrite $\boldsymbol{q}$ and $\boldsymbol{p}$ in terms of the principal stresses in the plane of deformation, σ_2 and σ_3, as

$$q = \frac{3\sqrt{3}}{2\sqrt{9 - \tan^2\psi}}(\sigma_2 - \sigma_3),$$

and

$$p = -\frac{1}{2}(\sigma_2 + \sigma_3) + \frac{\tan\psi}{2\sqrt{3(9 - \tan^2\psi)}}(\sigma_2 - \sigma_3).$$

With these expressions the Drucker-Prager yield surface can be written in terms of σ_2 and σ_3 as

$$\frac{9 - \tan\beta\tan\psi}{2\sqrt{3(9 - \tan^2\psi)}}(\sigma_2 - \sigma_3) + \frac{1}{2}\tan\beta(\sigma_2 + \sigma_3) - d = 0.$$

The Mohr-Coulomb yield surface in the $(2, 3)$ plane is

$$\sigma_2 - \sigma_3 + \sin\phi(\sigma_2 + \sigma_3) - 2c\cos\phi = 0.$$

By comparison,

$$\sin\phi = \frac{\tan\beta\sqrt{3(9 - \tan^2\psi)}}{9 - \tan\beta\tan\psi},$$

$$c\cos\phi = \frac{\sqrt{3(9-\tan^2\psi)}}{9-\tan\beta\tan\psi}\,d.$$

These relationships provide a match between the Mohr-Coulomb material parameters and linear Drucker-Prager material parameters in plane strain. Consider the two extreme cases of flow definition: associated flow, $\psi = \beta$, and nondilatant flow, when $\psi = 0$. For associated flow

$$\tan\beta = \frac{\sqrt{3}\sin\phi}{\sqrt{1+\frac{1}{3}\sin^2\phi}} \quad \text{and} \quad \frac{d}{c} = \frac{\sqrt{3}\cos\phi}{\sqrt{1+\frac{1}{3}\sin^2\phi}},$$

and for nondilatant flow

$$\tan\beta = \sqrt{3}\sin\phi \quad \text{and} \quad \frac{d}{c} = \sqrt{3}\cos\phi.$$

In either case σ_c^0 is immediately available as

$$\sigma_c^0 = \frac{1}{1-\frac{1}{3}\tan\beta}\,d.$$

The difference between these two approaches increases with the friction angle; however, the results are not very different for typical friction angles, as illustrated in Table 23.3.1–1.

Table 23.3.1–1 Plane strain matching of Drucker-Prager and Mohr-Coulomb models.

Mohr-Coulomb friction angle, ϕ	**Associated flow**		**Nondilatant flow**	
	Drucker-Prager friction angle, β	d/c	**Drucker-Prager friction angle, β**	d/c
10°	16.7°	1.70	16.7°	1.70
20°	30.2°	1.60	30.6°	1.63
30°	39.8°	1.44	40.9°	1.50
40°	46.2°	1.24	48.1°	1.33
50°	50.5°	1.02	53.0°	1.11

"Limit load calculations with granular materials," Section 1.15.4 of the Abaqus Benchmarks Manual, and "Finite deformation of an elastic-plastic granular material," Section 1.15.5 of the Abaqus Benchmarks Manual, show a comparison of the response of a simple loading of a granular material using the Drucker-Prager and Mohr-Coulomb models, using the plane strain approach to match the parameters of the two models.

Matching triaxial test response

Another approach to matching Mohr-Coulomb and Drucker-Prager model parameters for materials with low friction angles is to make the two models provide the same failure definition in triaxial compression and tension. The following matching procedure is applicable only to the linear Drucker-Prager model since this is the only model in this class that allows for different yield values in triaxial compression and tension.

We can rewrite the Mohr-Coulomb model in terms of principal stresses:

$$\sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - 2c\cos\phi = 0.$$

Using the results above for the stress invariants $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$ in triaxial compression and tension allows the linear Drucker-Prager model to be written for triaxial compression as

$$\sigma_1 - \sigma_3 + \frac{\tan\beta}{2 + \frac{1}{3}\tan\beta}(\sigma_1 + \sigma_3) - \frac{1 - \frac{1}{3}\tan\beta}{1 + \frac{1}{6}\tan\beta}\sigma_c^0 = 0,$$

and for triaxial tension as

$$\sigma_1 - \sigma_3 + \frac{\tan\beta}{\frac{2}{K} - \frac{1}{3}\tan\beta}(\sigma_1 + \sigma_3) - \frac{1 - \frac{1}{3}\tan\beta}{\frac{1}{K} - \frac{1}{6}\tan\beta}\sigma_c^0 = 0.$$

We wish to make these expressions identical to the Mohr-Coulomb model for all values of (σ_1, σ_3). This is possible by setting

$$K = \frac{1}{1 + \frac{1}{3}\tan\beta}.$$

By comparing the Mohr-Coulomb model with the linear Drucker-Prager model,

$$\tan\beta = \frac{6\sin\phi}{3 - \sin\phi},$$

$$\sigma_c^0 = 2c\frac{\cos\phi}{1 - \sin\phi},$$

and, hence, from the previous result

$$K = \frac{3 - \sin\phi}{3 + \sin\phi}.$$

These results for β, K, and σ_c^0 provide linear Drucker-Prager parameters that match the Mohr-Coulomb model in triaxial compression and tension.

The value of $\boldsymbol{K}$ in the linear Drucker-Prager model is restricted to $K \geq 0.778$ for the yield surface to remain convex. The result for $\boldsymbol{K}$ shows that this implies $\phi \leq 22°$. Many real materials have a larger Mohr-Coulomb friction angle than this value. One approach in such circumstances is to choose $K =$

0.778 and then to use the remaining equations to define β and σ_c^0. This approach matches the models for triaxial compression only, while providing the closest approximation that the model can provide to failure being independent of the intermediate principal stress. If ϕ is significantly larger than 22°, this approach may provide a poor Drucker-Prager match of the Mohr-Coulomb parameters. Therefore, this matching procedure is not generally recommended; use the Mohr-Coulomb model instead.

While using one-element tests to verify the calibration of the model, it should be noted that the Abaqus output variables SP1, SP2, and SP3 correspond to the principal stresses σ_3, σ_2, and σ_1, respectively.

Creep models for the linear Drucker-Prager model

Classical "creep" behavior of materials that exhibit plasticity according to the extended Drucker-Prager models can be defined in Abaqus/Standard. The creep behavior in such materials is intimately tied to the plasticity behavior (through the definitions of creep flow potentials and definitions of test data), so Drucker-Prager plasticity and Drucker-Prager hardening must be included in the material definition.

Creep and plasticity can be active simultaneously, in which case the resulting equations are solved in a coupled manner. To model creep only (without rate-independent plastic deformation), large values for the yield stress should be provided in the Drucker-Prager hardening definition: the result is that the material follows the Drucker-Prager model while it creeps, without ever yielding. When using this technique, a value must also be defined for the eccentricity, since, as described below, both the initial yield stress and eccentricity affect the creep potentials. This capability is limited to the linear model with a von Mises (circular) section in the deviatoric stress plane ($K = 1$; i.e., no third stress invariant effects are taken into account) and can be combined only with linear elasticity.

Creep behavior defined by the extended Drucker-Prager model is active only during soils consolidation, coupled temperature-displacement, and transient quasi-static procedures.

Creep formulation

The creep potential is hyperbolic, similar to the plastic flow potentials used in the hyperbolic and general exponent plasticity models. If creep properties are defined in Abaqus/Standard, the linear Drucker-Prager plasticity model also uses a hyperbolic plastic flow potential. As a consequence, if two analyses are run, one in which creep is not activated and another in which creep properties are specified but produce virtually no creep flow, the plasticity solutions will not be exactly the same: the solution with creep not activated uses a linear plastic potential, whereas the solution with creep activated uses a hyperbolic plastic potential.

Equivalent creep surface and equivalent creep stress

We adopt the notion of the existence of creep isosurfaces of stress points that share the same creep "intensity," as measured by an equivalent creep stress. When the material plastifies, it is desirable to have the equivalent creep surface coincide with the yield surface; therefore, we define the equivalent creep surfaces by homogeneously scaling down the yield surface. In the p–q plane that translates into parallels to the yield surface, as depicted in Figure 23.3.1–13. Abaqus/Standard requires that creep properties be described in terms of the same type of data used to define work hardening properties. The equivalent creep stress, $\bar{\sigma}^{cr}$, is then determined as follows:

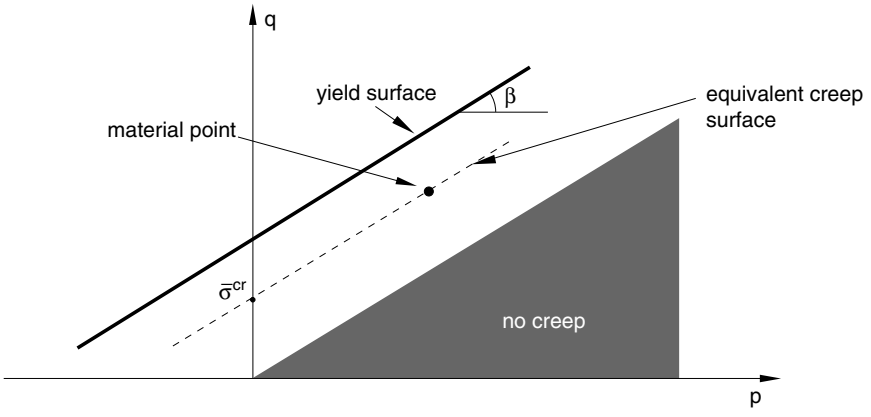


Figure 23.3.1–13 Equivalent creep stress defined as the shear stress.

$$\bar{\sigma}^{cr} = \frac{(q - p\tan\beta)}{(1 - \frac{1}{3}\tan\beta)} \quad \text{if creep is defined in terms of the uniaxial compression stress, } \sigma_c;$$
$$= \frac{(q - p\tan\beta)}{(1 + \frac{1}{3}\tan\beta)} \quad \text{if creep is defined in terms of the uniaxial tension stress, } \sigma_t;$$
$$= (q - p\tan\beta) \quad \text{if creep is defined in terms of the cohesion, } d.$$

Figure 23.3.1–13 shows how the equivalent point is determined when the material properties are in shear, with stress d. A consequence of these concepts is that there is a cone in p–q space inside which creep is not active since any point inside this cone would have a negative equivalent creep stress.

Creep flow

The creep strain rate in Abaqus/Standard is assumed to follow from the same hyperbolic potential as the plastic strain rate (see Figure 23.3.1–6):

$$G^{cr} = \sqrt{(\epsilon\bar{\sigma}|_0 \tan\psi)^2 + q^2} - p\tan\psi,$$

where

$\psi(\theta, f_i)$ is the dilation angle measured in the p–q plane at high confining pressure;

$\bar{\sigma}|_0 = \bar{\sigma}|_{\bar{\varepsilon}^{pl}=0, \dot{\bar{\varepsilon}}^{pl}=0}$ is the initial yield stress taken from the user-specified Drucker-Prager hardening data; and

ϵ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the creep potential tends to a straight line as the eccentricity tends to zero).

Suitable default values are provided for ϵ, as described below. This creep potential, which is continuous and smooth, ensures that the creep flow direction is always uniquely defined. The function approaches the linear Drucker-Prager flow potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane was shown in Figure 23.3.1–6. The creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

The default creep potential eccentricity is $\epsilon = 0.1$, which implies that the material has almost the same dilation angle over a wide range of confining pressure stress values. Increasing the value of ϵ provides more curvature to the creep potential, implying that the dilation angle increases as the confining pressure decreases. Values of ϵ that are significantly less than the default value may lead to convergence problems if the material is subjected to low confining pressures, because of the very tight curvature of the creep potential locally where it intersects the p-axis. For details on the behavior of these models refer to "Verification of creep integration," Section 3.2.6 of the Abaqus Benchmarks Manual.

If the creep material properties are defined by a compression test, numerical problems may arise for very low stress values. Abaqus/Standard protects for such a case, as described in "Models for granular or polymer behavior," Section 4.4.2 of the Abaqus Theory Manual.

Nonassociated flow

The use of a creep potential different from the equivalent creep surface implies that the material stiffness matrix is not symmetric; therefore, the unsymmetric matrix storage and solution scheme should be used (see "Defining an analysis," Section 6.1.2). If the difference between β and ψ is not large and the region of the model in which inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence and the unsymmetric matrix scheme may not be needed.

Specifying a creep law

The definition of creep behavior in Abaqus/Standard is completed by specifying the equivalent "uniaxial behavior"—the creep "law." In many practical cases the creep "law" is defined through user subroutine **CREEP** because creep laws are usually of very complex form to fit experimental data. Data input methods are provided for some simple cases, including two forms of a power law model and a variation of the Singh-Mitchell law.

User subroutine CREEP

User subroutine **CREEP** provides a very general capability for implementing viscoplastic models in Abaqus/Standard in which the strain rate potential can be written as a function of the equivalent stress and any number of "solution-dependent state variables." When used in conjunction with these material models, the equivalent creep stress, $\bar{\sigma}^{cr}$, is made available in the routine. Solution-dependent state variables are any variables that are used in conjunction with the constitutive definition and whose values evolve with the solution. Examples are hardening variables associated with the model. When a more general form is required for the stress potential, user subroutine **UMAT** can be used.

Input File Usage: *DRUCKER PRAGER CREEP, LAW=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Creep**: **Law: User**

"Time hardening" form of the power law model

The "time hardening" form of the power law model is

$$\dot{\bar{\varepsilon}}^{cr} = A(\bar{\sigma}^{cr})^n t^m,$$

where

$\dot{\bar{\varepsilon}}^{cr}$ is the equivalent creep strain rate, defined so that $\dot{\bar{\varepsilon}}^{cr} = |\dot{\varepsilon}^{cr}_{11}|$ if the equivalent creep stress is defined in uniaxial compression, $\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}^{cr}_{11}$ if defined in uniaxial tension, and $\dot{\bar{\varepsilon}}^{cr} = \dot{\gamma}^{cr}/\sqrt{3}$ if defined in pure shear, where γ^{cr} is the engineering shear creep strain;

$\bar{\sigma}^{cr}$ is the equivalent creep stress;

t is the total time; and

A, n, and m are user-defined creep material parameters specified as functions of temperature and field variables.

Input File Usage: *DRUCKER PRAGER CREEP, LAW=TIME

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Creep**: **Law: Time**

"Strain hardening" form of the power law model

As an alternative to the "time hardening" form of the power law, as defined above, the corresponding "strain hardening" form can be used:

$$\dot{\bar{\varepsilon}}^{cr} = \left(A(\bar{\sigma}^{cr})^n [(m+1)\bar{\varepsilon}^{cr}]^m \right)^{\frac{1}{m+1}}.$$

For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$.

Input File Usage: *DRUCKER PRAGER CREEP, LAW=STRAIN

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Creep**: **Law: Strain**

Singh-Mitchell law

A second creep law available as data input is a variation of the Singh-Mitchell law:

$$\dot{\bar{\varepsilon}}^{cr} = Ae^{(\alpha\bar{\sigma}^{cr})}(t_1/t)^m,$$

where $\dot{\bar{\varepsilon}}^{cr}$, t, and $\bar{\sigma}^{cr}$ are defined above and A, α, t_1, and m are user-defined creep material parameters specified as functions of temperature and field variables. For physically reasonable behavior A and α must be positive, $0.0 < m \leq 1.0$, and t_1 should be small compared to the total time.

Input File Usage: *DRUCKER PRAGER CREEP, LAW=SINGHM

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Drucker Prager**: **Suboptions→Drucker Prager Creep**: **Law: SinghM**

Numerical difficulties

Depending on the choice of units for the creep laws described above, the value of A may be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations; therefore, use another system of units to avoid such difficulties in the calculation of creep strain increments.

Creep integration

Abaqus/Standard provides both explicit and implicit time integration of creep and swelling behavior. The choice of the time integration scheme depends on the procedure type, the parameters specified for the procedure, the presence of plasticity, and whether or not a geometric linear or nonlinear analysis is requested, as discussed in "Rate-dependent plasticity: creep and swelling," Section 23.2.4.

Initial conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, Abaqus allows you to prescribe initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, by specifying the conditions directly (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). For more complicated cases initial conditions can be defined in Abaqus/Standard through user subroutine **HARDINI**.

Input File Usage: Use the following option to specify the initial equivalent plastic strain directly:

*INITIAL CONDITIONS, TYPE=HARDENING

Use the following option in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine **HARDINI**:

*INITIAL CONDITIONS, TYPE=HARDENING, USER

Abaqus/CAE Usage: Use the following options to specify the initial equivalent plastic strain directly:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use the following options in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine **HARDINI**:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

The Drucker-Prager models can be used with the following element types: plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements. All Drucker-Prager models are

also available in plane stress (plane stress, shell, and membrane elements), except for the linear Drucker-Prager model with creep.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning for the Drucker-Prager plasticity/creep model:

PEEQ — Equivalent plastic strain.

For the linear Drucker-Prager plasticity model PEEQ is defined as $\bar{\varepsilon}^{pl}|_0 + \int_0^t \dot{\varepsilon}^{pl} dt$; where $\bar{\varepsilon}^{pl}|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions") and $\dot{\varepsilon}^{pl}$ is the equivalent plastic strain rate.

For the hyperbolic and exponential Drucker-Prager plasticity models PEEQ is defined as $\bar{\varepsilon}^{pl}|_0 + \int \frac{\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}^{pl}}{\sigma^0}$, where $\bar{\varepsilon}^{pl}|_0$ is the initial equivalent plastic strain and σ^0 is the yield stress.

CEEQ — Equivalent creep strain, $\int \dot{\bar{\varepsilon}}^{cr} dt$.

23.3.2 MODIFIED DRUCKER-PRAGER/CAP MODEL

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Inelastic behavior," Section 23.1.1
- "Material library: overview," Section 21.1.1
- "Rate-dependent plasticity: creep and swelling," Section 23.2.4
- "CREEP," Section 1.1.1 of the Abaqus User Subroutines Reference Manual
- *CAP PLASTICITY
- *CAP HARDENING
- *CAP CREEP
- "Defining cap plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The modified Drucker-Prager/Cap plasticity/creep model:

- is intended to model cohesive geological materials that exhibit pressure-dependent yield, such as soils and rocks;
- is based on the addition of a cap yield surface to the Drucker-Prager plasticity model ("Extended Drucker-Prager models," Section 23.3.1), which provides an inelastic hardening mechanism to account for plastic compaction and helps to control volume dilatancy when the material yields in shear;
- can be used in Abaqus/Standard to simulate creep in materials exhibiting long-term inelastic deformation through a cohesion creep mechanism in the shear failure region and a consolidation creep mechanism in the cap region;
- can be used in conjunction with either the elastic material model ("Linear elastic behavior," Section 22.2.1) or, in Abaqus/Standard if creep is not defined, the porous elastic material model ("Elastic behavior of porous materials," Section 22.3.1); and
- provides a reasonable response to large stress reversals in the cap region; however, in the failure surface region the response is reasonable only for essentially monotonic loading.

Yield surface

The addition of the cap yield surface to the Drucker-Prager model serves two main purposes: it bounds the yield surface in hydrostatic compression, thus providing an inelastic hardening mechanism to represent plastic compaction; and it helps to control volume dilatancy when the material yields in shear

by providing softening as a function of the inelastic volume increase created as the material yields on the Drucker-Prager shear failure surface.

The yield surface has two principal segments: a pressure-dependent Drucker-Prager shear failure segment and a compression cap segment, as shown in Figure 23.3.2–1. The Drucker-Prager failure segment is a perfectly plastic yield surface (no hardening). Plastic flow on this segment produces inelastic volume increase (dilation) that causes the cap to soften. On the cap surface plastic flow causes the material to compact. The model is described in detail in "Drucker-Prager/Cap model for geological materials," Section 4.4.4 of the Abaqus Theory Manual.

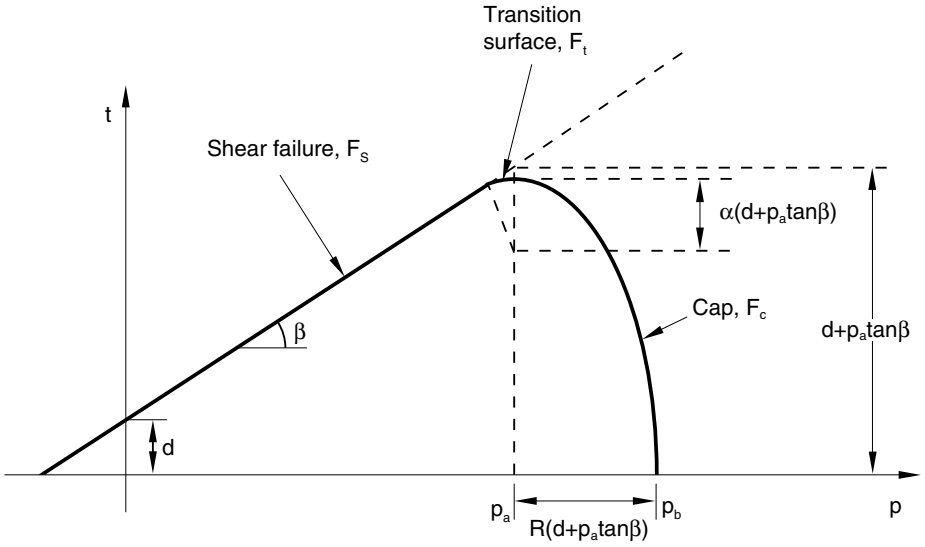


Figure 23.3.2–1 Modified Drucker-Prager/Cap model: yield surfaces in the p–t plane.

Failure surface

The Drucker-Prager failure surface is written as

$$F_s = t - p\tan\beta - d = 0,$$

where $\beta(\theta, f_i)$ and $d(\theta, f_i)$ represent the angle of friction of the material and its cohesion, respectively, and can depend on temperature, θ, and other predefined fields $f_i, i = 1, 2, 3 \ldots$. The deviatoric stress measure t is defined as

$$t = \frac{1}{2}q\left[1 + \frac{1}{K} - \left(1 - \frac{1}{K}\right)\left(\frac{r}{q}\right)^3\right];$$

and

$p = -\frac{1}{3}\text{trace}\,(\boldsymbol{\sigma})$ is the equivalent pressure stress,

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$ is the Mises equivalent stress,

$r = \left(\frac{9}{2}\mathbf{S} : \mathbf{S} \cdot \mathbf{S}\right)^{\frac{1}{3}}$ is the third stress invariant, and

$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$ is the deviatoric stress.

$K(\theta, f_i)$ is a material parameter that controls the dependence of the yield surface on the value of the intermediate principal stress, as shown in Figure 23.3.2–2.

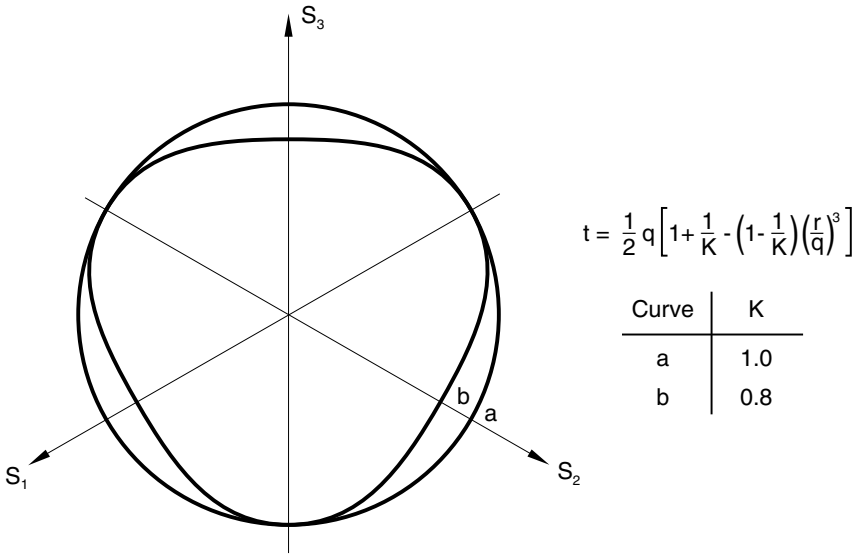


Figure 23.3.2–2 Typical yield/flow surfaces in the deviatoric plane.

The yield surface is defined so that K is the ratio of the yield stress in triaxial tension to the yield stress in triaxial compression. $K = 1$ implies that the yield surface is the von Mises circle in the deviatoric principal stress plane (the Π-plane), so that the yield stresses in triaxial tension and compression are the same; this is the default behavior in Abaqus/Standard and the only behavior available in Abaqus/Explicit. To ensure that the yield surface remains convex requires $0.778 \leq K \leq 1.0$.

Cap yield surface

The cap yield surface has an elliptical shape with constant eccentricity in the meridional (p–t) plane (Figure 23.3.2–1) and also includes dependence on the third stress invariant in the deviatoric plane (Figure 23.3.2–2). The cap surface hardens or softens as a function of the volumetric inelastic strain: volumetric plastic and/or creep compaction (when yielding on the cap and/or creeping according to the consolidation mechanism, as described later in this section) causes hardening, while volumetric

plastic and/or creep dilation (when yielding on the shear failure surface and/or creeping according to the cohesion mechanism, as described later in this section) causes softening. The cap yield surface is

$$F_c = \sqrt{[p - p_a]^2 + \left[\frac{Rt}{(1 + \alpha - \alpha/\cos\beta)}\right]^2} - R(d + p_a \tan\beta) = 0,$$

where $R(\theta, f_i)$ is a material parameter that controls the shape of the cap, $\alpha(\theta, f_i)$ is a small number that we discuss later, and $p_a(\varepsilon^{pl}_{\text{vol}} + \varepsilon^{cr}_{\text{vol}})$ is an evolution parameter that represents the volumetric inelastic strain driven hardening/softening. The hardening/softening law is a user-defined piecewise linear function relating the hydrostatic compression yield stress, p_b, and volumetric inelastic strain (Figure 23.3.2–3):

$$p_b = p_b(\varepsilon^{in}_{\text{vol}}|_0 + \varepsilon^{pl}_{\text{vol}} + \varepsilon^{cr}_{\text{vol}}).$$

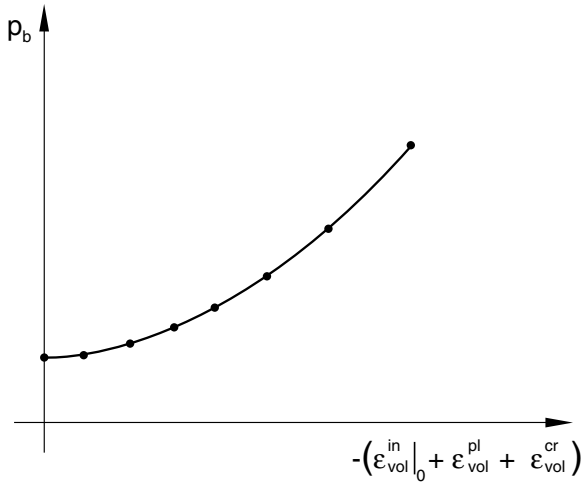


Figure 23.3.2–3 Typical Cap hardening.

The volumetric inelastic strain axis in Figure 23.3.2–3 has an arbitrary origin: $\varepsilon^{in}_{\text{vol}}|_0$ $(= \varepsilon^{pl}_{\text{vol}}|_0 + \varepsilon^{cr}_{\text{vol}}|_0)$ is the position on this axis corresponding to the initial state of the material when the analysis begins, thus defining the position of the cap (p_b) in Figure 23.3.2–1 at the start of the analysis. The evolution parameter p_a is given as

$$p_a = \frac{p_b - Rd}{(1 + R\tan\beta)}.$$

The parameter α is a small number (typically 0.01 to 0.05) used to define a transition yield surface,

$$F_t = \sqrt{[p-p_a]^2 + \left[t - (1 - \frac{\alpha}{\cos\beta})(d + p_a \tan\beta)\right]^2} - \alpha(d + p_a \tan\beta) = 0,$$

so that the model provides a smooth intersection between the cap and failure surfaces.

Defining yield surface variables

You provide the variables $\boldsymbol{d}$, β, $\boldsymbol{R}$, $\varepsilon^{in}_{vol}|_0$, α, and $\boldsymbol{K}$ to define the shape of the yield surface. In Abaqus/Standard $0.778 \leq K \leq 1.0$, while in Abaqus/Explicit $\boldsymbol{K} = 1$ ($t = q$). If desired, combinations of these variables can also be defined as a tabular function of temperature and other predefined field variables.

Input File Usage: *CAP PLASTICITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Cap Plasticity**

Defining hardening parameters

The hardening curve specified for this model interprets yielding in the hydrostatic pressure sense: the hydrostatic pressure yield stress is defined as a tabular function of the volumetric inelastic strain, and, if desired, a function of temperature and other predefined field variables. The range of values for which p_b is defined should be sufficient to include all values of effective pressure stress that the material will be subjected to during the analysis.

Input File Usage: *CAP HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Cap Plasticity**: **Suboptions→Cap Hardening**

Plastic flow

Plastic flow is defined by a flow potential that is associated in the deviatoric plane, associated in the cap region in the meridional plane, and nonassociated in the failure surface and transition regions in the meridional plane. The flow potential surface that we use in the meridional plane is shown in Figure 23.3.2–4: it is made up of an elliptical portion in the cap region that is identical to the cap yield surface,

$$G_c = \sqrt{[p-p_a]^2 + \left[\frac{Rt}{(1 + \alpha - \alpha/\cos\beta)}\right]^2},$$

and another elliptical portion in the failure and transition regions that provides the nonassociated flow component in the model,

$$G_s = \sqrt{[(p_a - p)\tan\beta]^2 + \left[\frac{t}{(1 + \alpha - \alpha/\cos\beta)}\right]^2}.$$

The two elliptical portions form a continuous and smooth potential surface.

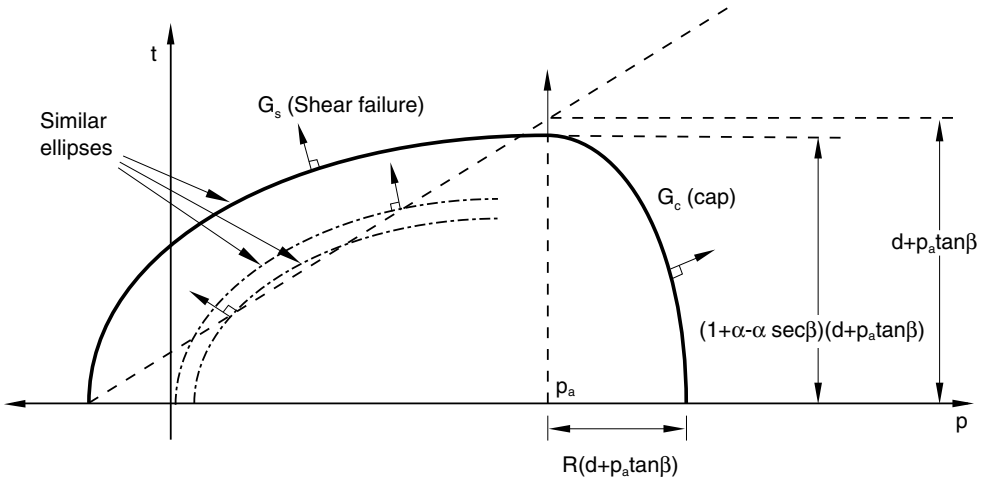


Figure 23.3.2–4 Modified Drucker-Prager/Cap model: flow potential in the p–t plane.

Nonassociated flow

Nonassociated flow implies that the material stiffness matrix is not symmetric and the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see "Defining an analysis," Section 6.1.2). If the region of the model in which nonassociated inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence; in such cases the unsymmetric matrix scheme may not be needed.

Calibration

At least three experiments are required to calibrate the simplest version of the Cap model: a hydrostatic compression test (an oedometer test is also acceptable) and either two triaxial compression tests or one triaxial compression test and one uniaxial compression test (more than two tests are recommended for a more accurate calibration).

The hydrostatic compression test is performed by pressurizing the sample equally in all directions. The applied pressure and the volume change are recorded.

The uniaxial compression test involves compressing the sample between two rigid platens. The load and displacement in the direction of loading are recorded. The lateral displacements should also be recorded so that the correct volume changes can be calibrated.

Triaxial compression experiments are performed using a standard triaxial machine where a fixed confining pressure is maintained while the differential stress is applied. Several tests covering the range of confining pressures of interest are usually performed. Again, the stress and strain in the direction of loading are recorded, together with the lateral strain so that the correct volume changes can be calibrated.

Unloading measurements in these tests are useful to calibrate the elasticity, particularly in cases where the initial elastic region is not well defined.

The hydrostatic compression test stress-strain curve gives the evolution of the hydrostatic compression yield stress, $p_b(\varepsilon_{vol}^{pl})$, required for the cap hardening curve definition.

The friction angle, β, and cohesion, d, which define the shear failure dependence on hydrostatic pressure, are calculated by plotting the failure stresses of the two triaxial compression tests (or the triaxial compression test and the uniaxial compression test) in the pressure stress (p) versus shear stress (q) space: the slope of the straight line passing through the two points gives the angle β and the intersection with the q-axis gives d. For more details on the calibration of β and d, see the discussion on calibration in "Extended Drucker-Prager models," Section 23.3.1.

R represents the curvature of the cap part of the yield surface and can be calibrated from a number of triaxial tests at high confining pressures (in the cap region). R must be between 0.0001 and 1000.0.

Abaqus/Standard creep model

Classical "creep" behavior of materials that exhibit plasticity according to the capped Drucker-Prager plasticity model can be defined in Abaqus/Standard. The creep behavior in such materials is intimately tied to the plasticity behavior (through the definitions of creep flow potentials and definitions of test data), so cap plasticity and cap hardening must be included in the material definition. If no rate-independent plastic behavior is desired in the model, large values for the cohesion, d, as well as large values for the compression yield stress, p_b, should be provided in the plasticity definition: as a result the material follows the capped Drucker-Prager model while it creeps, without ever yielding. This capability is limited to cases in which there is no third stress invariant dependence of the yield surface ($K = 1$) and cases in which the yield surface has no transition region ($\alpha = 0$). The elastic behavior must be defined using linear isotropic elasticity (see "Defining isotropic elasticity" in "Linear elastic behavior," Section 22.2.1).

Creep behavior defined for the modified Drucker-Prager/Cap model is active only during soils consolidation, coupled temperature-displacement, and transient quasi-static procedures.

Creep formulation

This model has two possible creep mechanisms that are active in different loading regions: one is a cohesion mechanism, which follows the type of plasticity active in the shear-failure plasticity region, and the other is a consolidation mechanism, which follows the type of plasticity active in the cap plasticity region. Figure 23.3.2–5 shows the regions of applicability of the creep mechanisms in p–q space.

Equivalent creep surface and equivalent creep stress for the cohesion creep mechanism

Consider the cohesion creep mechanism first. We adopt the notion of the existence of creep isosurfaces of stress points that share the same creep "intensity," as measured by an equivalent creep stress. Since it is desirable to have the equivalent creep surface coincide with the yield surface, we define the equivalent creep surfaces by homogeneously scaling down the yield surface. In the p–q plane the equivalent creep surfaces translate into surfaces that are parallel to the yield surface, as depicted in Figure 23.3.2–6.

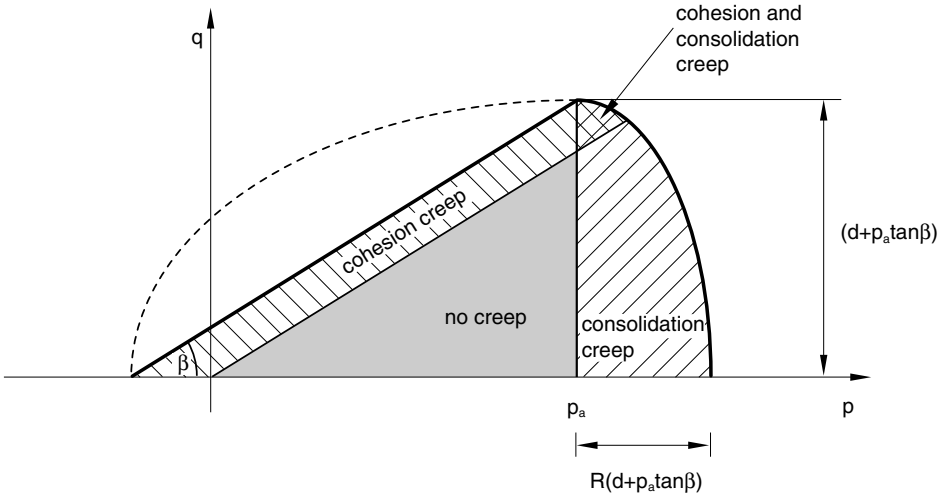


Figure 23.3.2–5 Regions of activity of creep mechanisms.

Abaqus/Standard requires that cohesion creep properties be measured in a uniaxial compression test. The equivalent creep stress, $\bar{\sigma}^{cr}$, is determined as follows:

$$\bar{\sigma}^{cr} = \frac{(q - p\tan\beta)}{(1 - \frac{1}{3}\tan\beta)} .$$

Abaqus/Standard also requires that $\bar{\sigma}^{cr}$ be positive. Figure 23.3.2–6 shows such an equivalent creep stress. A consequence of these concepts is that there is a cone in p–q space inside which creep is not active. Any point inside this cone would have a negative equivalent creep stress.

Equivalent creep surface and equivalent creep stress for the consolidation creep mechanism

Next, consider the consolidation creep mechanism. In this case we wish to make creep dependent on the hydrostatic pressure above a threshold value of p_a, with a smooth transition to the areas in which the mechanism is not active ($p < p_a$). Therefore, we define equivalent creep surfaces as constant hydrostatic pressure surfaces (vertical lines in the p–q plane). Abaqus/Standard requires that consolidation creep properties be measured in a hydrostatic compression test. The effective creep pressure, $\bar{p}^{cr}$, is then the point on the p-axis with a relative pressure of $\bar{p}^{cr} = p - p_a$. This value is used in the uniaxial creep law. The equivalent volumetric creep strain rate produced by this type of law is defined as positive for a positive equivalent pressure. The internal tensor calculations in Abaqus/Standard account for the fact that a positive pressure will produce negative (that is, compressive) volumetric creep components.

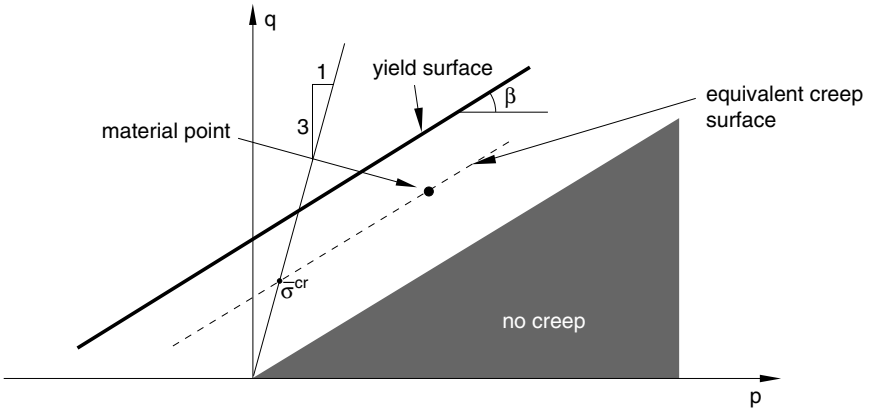


Figure 23.3.2–6 Equivalent creep stress for cohesion creep.

Creep flow

The creep strain rate produced by the cohesion mechanism is assumed to follow a potential that is similar to that of the creep strain rate in the Drucker-Prager creep model ("Extended Drucker-Prager models," Section 23.3.1); that is, a hyperbolic function:

$$G_s^{cr} = \sqrt{(0.1\frac{d}{(1-\frac{1}{3}\tan\beta)}\tan\beta)^2 + q^2} - p\tan\beta.$$

This creep flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches a parallel to the shear-failure yield surface asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at a right angle. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 23.3.2–7. The cohesion creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

Abaqus/Standard protects for numerical problems that may arise for very low stress values. See "Drucker-Prager/Cap model for geological materials," Section 4.4.4 of the Abaqus Theory Manual, for details.

The creep strain rate produced by the consolidation mechanism is assumed to follow a potential that is similar to that of the plastic strain rate in the cap yield surface (Figure 23.3.2–8):

$$G_c^{cr} = \sqrt{[p-p_a]^2 + [Rq]^2}.$$

The consolidation creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane). The volumetric components of creep strain from both mechanisms contribute to the hardening/softening of the cap, as described previously. For details on the behavior of these models refer to "Verification of creep integration," Section 3.2.6 of the Abaqus Benchmarks Manual.

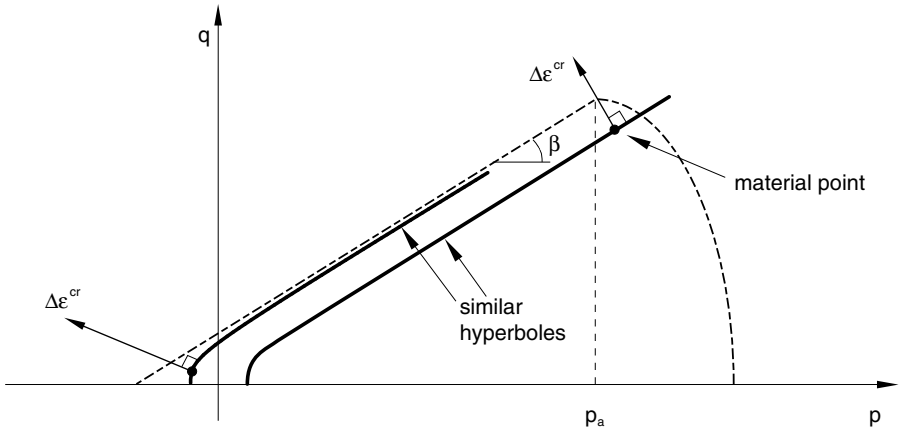


Figure 23.3.2–7 Cohesion creep potentials in the p–q plane.

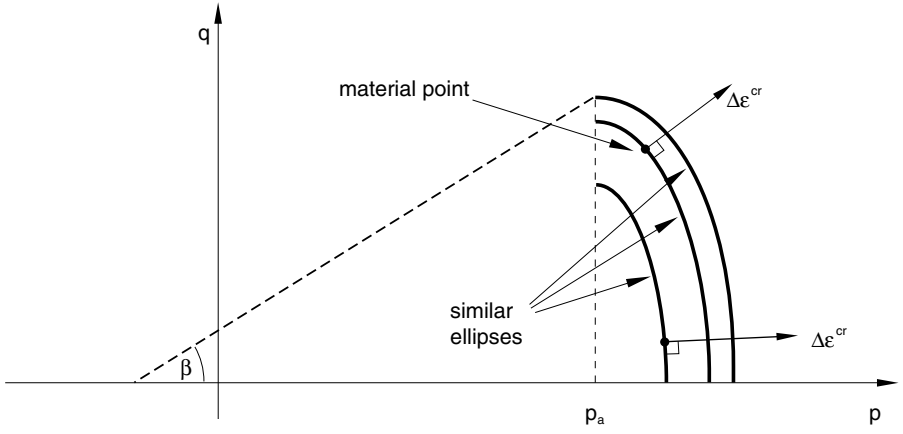


Figure 23.3.2–8 Consolidation creep potentials in the p–q plane.

Nonassociated flow

The use of a creep potential for the cohesion mechanism different from the equivalent creep surface implies that the material stiffness matrix is not symmetric, and the unsymmetric matrix storage and solution scheme should be used (see "Defining an analysis," Section 6.1.2). If the region of the model in which cohesive inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence; in such cases the unsymmetric matrix scheme may not be needed.

Specifying creep laws

The definition of the creep behavior is completed by specifying the equivalent "uniaxial behavior"—the creep "laws." In many practical cases the creep laws are defined through user subroutine **CREEP** because creep laws are usually of complex form to fit experimental data. Data input methods are provided for some simple cases.

User subroutine CREEP

User subroutine **CREEP** provides a general capability for implementing viscoplastic models in which the strain rate potential can be written as a function of the equivalent stress and any number of "solution-dependent state variables." When used in conjunction with these materials, the equivalent cohesion creep stress, $\bar{\sigma}^{cr}$, and the effective creep pressure, $\bar{p}^{cr}$, are made available in the routine. Solution-dependent state variables are any variables that are used in conjunction with the constitutive definition and whose values evolve with the solution. Examples are hardening variables associated with the model. When a more general form is required for the stress potential, user subroutine **UMAT** can be used.

Input File Usage: Use either or both of the following options:

∗CAP CREEP, MECHANISM=COHESION, LAW=USER
∗CAP CREEP, MECHANISM=CONSOLIDATION, LAW=USER

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical→Plasticity→Cap Plasticity**:
Suboptions→Cap Creep Cohesion: **Law: User**
Suboptions→Cap Creep Consolidation: **Law: User**

"Time hardening" form of the power law model

With respect to the cohesion mechanism, the power law is available

$$\dot{\bar{\varepsilon}}^{cr} = A(\bar{\sigma}^{cr})^n t^m,$$

where

$\dot{\bar{\varepsilon}}^{cr}$	is the equivalent creep strain rate;
$\bar{\sigma}^{cr}$	is the equivalent cohesion creep stress;
t	is the total time; and
A, n, and m	are user-defined creep material parameters specified as functions of temperature and field variables.

In using this form of the power law model with the consolidation mechanism, $\bar{\sigma}^{cr}$ can be replaced by $\bar{p}^{cr}$, the effective creep pressure, in the above relation.

Input File Usage: Use either or both of the following options:

∗CAP CREEP, MECHANISM=COHESION, LAW=TIME
∗CAP CREEP, MECHANISM=CONSOLIDATION, LAW=TIME

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical→Plasticity→Cap Plasticity**:
Suboptions→Cap Creep Cohesion: **Law: Time**
Suboptions→Cap Creep Consolidation: **Law: Time**

"Strain hardening" form of the power law model

As an alternative to the "time hardening" form of the power law, as defined above, the corresponding "strain hardening" form can be used. For the cohesion mechanism this law has the form

$$\dot{\bar{\varepsilon}}^{cr} = \left(A(\bar{\sigma}^{cr})^n [(m+1)\bar{\varepsilon}^{cr}]^m \right)^{\frac{1}{m+1}}.$$

In using this form of the power law model with the consolidation mechanism, $\bar{\sigma}^{cr}$ can be replaced by $\bar{p}^{cr}$, the effective creep pressure, in the above relation.

For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$.

Input File Usage: Use either or both of the following options:

∗CAP CREEP, MECHANISM=COHESION, LAW=STRAIN
∗CAP CREEP, MECHANISM=CONSOLIDATION, LAW=STRAIN

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical→Plasticity→Cap Plasticity**:
Suboptions→Cap Creep Cohesion: **Law: Strain**
Suboptions→Cap Creep Consolidation: **Law: Strain**

Singh-Mitchell law

A second cohesion creep law available as data input is a variation of the Singh-Mitchell law:

$$\dot{\bar{\varepsilon}}^{cr} = Ae^{(\alpha\bar{\sigma}^{cr})}(t_1/t)^m,$$

where $\dot{\bar{\varepsilon}}^{cr}$, t, and $\bar{\sigma}^{cr}$ are defined above and A, α, t_1, and m are user-defined creep material parameters specified as functions of temperature and field variables. For physically reasonable behavior A and α must be positive, $0.0 < m \leq 1.0$, and t_1 should be small compared to the total time.

In using this variation of the Singh-Mitchell law with the consolidation mechanism, $\bar{\sigma}^{cr}$ can be replaced by $\bar{p}^{cr}$, the effective creep pressure, in the above relation.

Input File Usage: Use either or both of the following options:

∗CAP CREEP, MECHANISM=COHESION, LAW=SINGHM
∗CAP CREEP, MECHANISM=CONSOLIDATION, LAW=SINGHM

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical→Plasticity→Cap Plasticity**:
Suboptions→Cap Creep Cohesion: **Law: SinghM**
Suboptions→Cap Creep Consolidation: **Law: SinghM**

Numerical difficulties

Depending on the choice of units for the creep laws described above, the value of A may be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations; therefore, use another system of units to avoid such difficulties in the calculation of creep strain increments.

Creep integration

Abaqus/Standard provides both explicit and implicit time integration of creep and swelling behavior. The choice of the time integration scheme depends on the procedure type, the parameters specified for the procedure, the presence of plasticity, and whether or not a geometric linear or nonlinear analysis is requested, as discussed in "Rate-dependent plasticity: creep and swelling," Section 23.2.4.

Initial conditions

The initial stress at a point can be defined (see "Defining initial stresses" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). If such a stress point lies outside the initially defined cap or transition yield surfaces and under the projection of the shear failure surface in the p–t plane (illustrated in Figure 23.3.2–1), Abaqus will try to adjust the initial position of the cap to make the stress point lie on the yield surface and a warning message will be issued. If the stress point lies outside the Drucker-Prager failure surface (or above its projection), an error message will be issued and execution will be terminated.

Elements

The modified Drucker-Prager/Cap material behavior can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning in the cap plasticity/creep model:

PEEQ	p_b, the cap position.
PEQC	Equivalent plastic strains for all three possible yield/failure surfaces (Drucker-Prager failure surface - PEQC1, cap surface - PEQC2, and transition surface - PEQC3) and the total volumetric inelastic strain (PEQC4). For each yield/failure surface, the equivalent plastic strain is $\bar{\varepsilon}^{pl} = \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the corresponding rate of plastic flow. The total volumetric inelastic strain is defined as $\varepsilon^{in}_{\text{vol}} = \int_0^t \dot{\varepsilon}^{pl}_{kk} dt + \int_0^t \dot{\varepsilon}^{cr}_{kk} dt$.

CEEQ Equivalent creep strain produced by the cohesion creep mechanism, defined as $\int \frac{\boldsymbol{\sigma}:d\boldsymbol{\varepsilon}^{cr}}{\bar{\sigma}^{cr}}$, where $\bar{\sigma}^{cr} = \frac{(q - p\tan\beta)}{(1-\frac{1}{3}\tan\beta)}$ is the equivalent creep stress.

CESW Equivalent creep strain produced by the consolidation creep mechanism, defined as $\int \frac{\boldsymbol{\sigma}:d\boldsymbol{\varepsilon}^{cr}}{\bar{p}}$, where $\bar{p} = \frac{R^2\ q^2 + p\ (p-p_a)}{G_c^{cr}}$ is the equivalent creep pressure.

23.3.3 MOHR-COULOMB PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *MOHR COULOMB
- *MOHR COULOMB HARDENING
- *TENSION CUTOFF
- "Defining Mohr-Coulomb plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The Mohr-Coulomb plasticity model:

- is used to model materials with the classical Mohr-Coloumb yield criterion;
- allows the material to harden and/or soften isotropically;
- uses a smooth flow potential that has a hyperbolic shape in the meridional stress plane and a piecewise elliptic shape in the deviatoric stress plane;
- is used with the linear elastic material model ("Linear elastic behavior," Section 22.2.1);
- can be used with the Rankine surface (tension cutoff) to limit load carrying capacity near the tensile region; and
- can be used for design applications in the geotechnical engineering area to simulate material response under essentially monotonic loading.

Elastic behavior

The elastic part of the response is specified as described in "Linear elastic behavior," Section 22.2.1. Linear isotropic elasticity is assumed.

Plastic behavior: yield criteria

The yield surface is a composite of two different criteria: a shear criterion, known as the Mohr-Coulomb surface, and an optional tension cutoff criterion, modeled using the Rankine surface.

Mohr-Coulomb surface

The Mohr-Coulomb criterion assumes that yield occurs when the shear stress on any point in a material reaches a value that depends linearly on the normal stress in the same plane. The Mohr-Coulomb model is based on plotting Mohr's circle for states of stress at yield in the plane of the maximum and

minimum principal stresses. The yield line is the best straight line that touches these Mohr's circles (Figure 23.3.3–1).

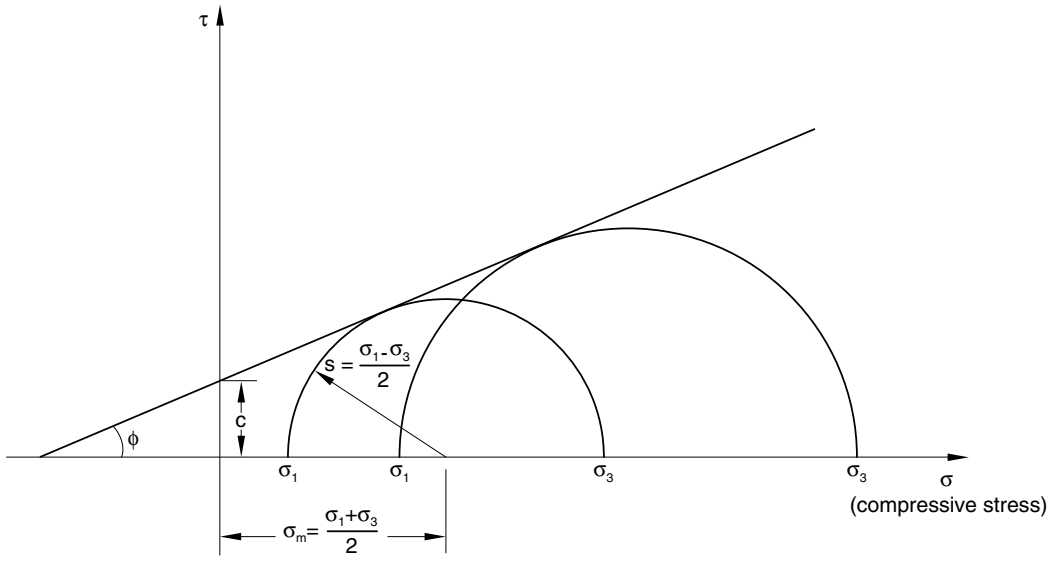


Figure 23.3.3–1 Mohr-Coulomb yield model.

Therefore, the Mohr-Coulomb model is defined by

$$\tau = c - \sigma \tan \phi,$$

where σ is negative in compression. From Mohr's circle,

$$\tau = s \cos \phi,$$
$$\sigma = \sigma_m + s \sin \phi.$$

Substituting for τ and σ, multiplying both sides by $\cos \phi$, and reducing, the Mohr-Coulomb model can be written as

$$s + \sigma_m \sin \phi - c \cos \phi = 0,$$

where

$$s = \frac{1}{2}(\sigma_1 - \sigma_3)$$

is half of the difference between the maximum principal stress, σ_1, and the minimum principal stress, σ_3 (and is, therefore, the maximum shear stress),

$$\sigma_m = \frac{1}{2}(\sigma_1 + \sigma_3)$$

is the average of the maximum and minimum principal stresses, and ϕ is the friction angle.

For general states of stress the model is more conveniently written in terms of three stress invariants as

$$F = R_{mc}q - p\tan\phi - c = 0,$$

where

$$R_{mc}(\Theta,\phi) = \frac{1}{\sqrt{3}\cos\phi}\sin\left(\Theta + \frac{\pi}{3}\right) + \frac{1}{3}\cos\left(\Theta + \frac{\pi}{3}\right)\tan\phi,$$

ϕ	is the slope of the Mohr-Coulomb yield surface in the p–$R_{mc}q$ stress plane (see Figure 23.3.3–2), which is commonly referred to as the friction angle of the material and can depend on temperature and predefined field variables;
c	is the cohesion of the material; and
Θ	is the deviatoric polar angle defined as

$$\cos(3\Theta) = \left(\frac{r}{q}\right)^3,$$

and

$p = -\frac{1}{3}\text{trace}\,(\boldsymbol{\sigma})$	is the equivalent pressure stress,
$q = \sqrt{\frac{3}{2}(\mathbf{S}:\mathbf{S})}$	is the Mises equivalent stress,
$r = \left(\frac{9}{2}\mathbf{S}\cdot\mathbf{S}:\mathbf{S}\right)^{\frac{1}{3}}$	is the third invariant of deviatoric stress, and
$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$	is the deviatoric stress.

The friction angle, ϕ, controls the shape of the yield surface in the deviatoric plane as shown in Figure 23.3.3–2. The tension cutoff surface is shown for a meridional angle of $\Theta = 0$. The friction angle range is $0° \le \phi < 90°$. In the case of $\phi = 0°$ the Mohr-Coulomb model reduces to the pressure-independent Tresca model with a perfectly hexagonal deviatoric section. In the case of $\phi = 90°$ the Mohr-Coulomb model reduces to the "tension cutoff" Rankine model with a triangular deviatoric section and $R_{mc} = \infty$ (this limiting case is not permitted within the Mohr-Coulomb model described here).

When using one-element tests to verify the calibration of the model, the output variables SP1, SP2, and SP3 correspond to the principal stresses σ_3, σ_2, and σ_1, respectively.

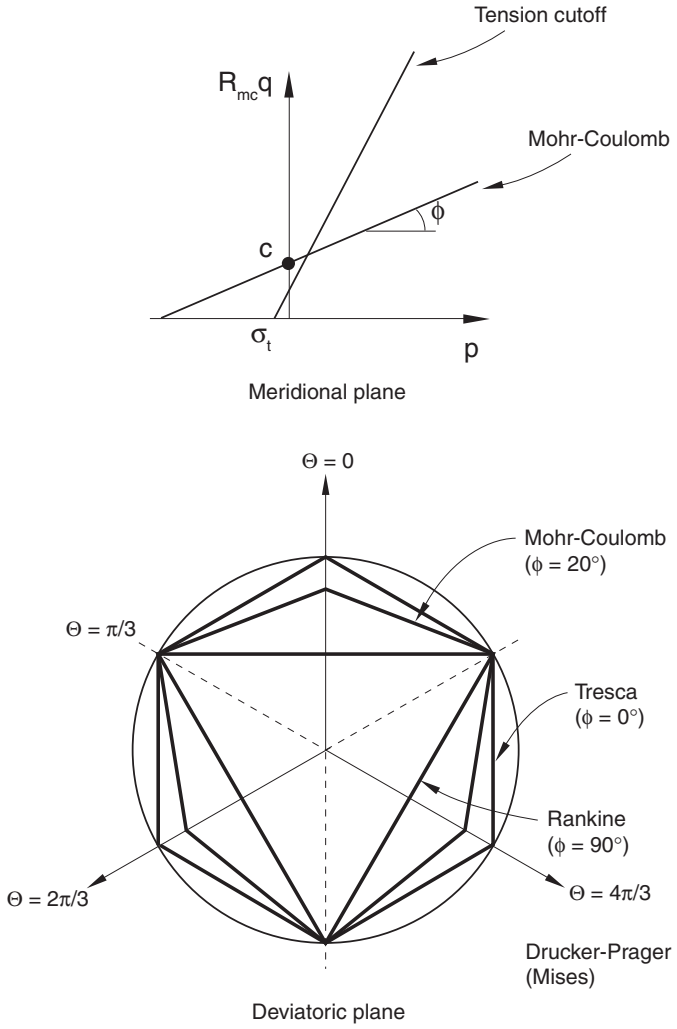


Figure 23.3.3–2 Mohr-Coulomb and tension cutoff surfaces in meridional and deviatoric planes.

Isotropic cohesion hardening is assumed for the hardening behavior of the Mohr-Coulomb yield surface. The hardening curve must describe the cohesion yield stress as a function of plastic strain and, possibly, temperature and predefined field variables. In defining this dependence at finite strains, "true" (Cauchy) stress and logarithmic strain values should be given. An optional tension cutoff hardening (or softening) curve can be specified

Rate dependency effects are not accounted for in this plasticity model.

Input File Usage: Use the following options to specify the Mohr-Coulomb yield surface and cohesion hardening:

*MOHR COULOMB

*MOHR COULOMB HARDENING

Abaqus/CAE Usage: Use the following options to specify the Mohr-Coulomb yield surface and cohesion hardening:

Property module: material editor: **Mechanical→Plasticity→Mohr Coulomb Plasticity**

Property module: material editor: **Mechanical→Plasticity→Mohr Coulomb Plasticity**: **Cohesion**

Rankine surface

In Abaqus tension cutoff is modeled using the Rankine surface, which is written as

$$F_t = R_r(\Theta)q - p - \sigma_t(\bar{\varepsilon}_t^{pl}) = 0,$$

where $R_r(\Theta) = (2/3)\cos\Theta$, and σ_t is the tension cutoff value representing softening (or hardening) of the Rankine surface, as a function of tensile equivalent plastic strain, $\bar{\varepsilon}_t^{pl}$.

Input File Usage: Use the following option to specify hardening or softening for the Rankine surface:

*TENSION CUTOFF

Abaqus/CAE Usage: Use the following option to specify hardening or softening for the Rankine surface:

Property module: material editor: **Mechanical→Plasticity→Mohr Coulomb Plasticity**: toggle on **Specify tension cutoff**; **Tension Cutoff**

Plastic behavior: flow potentials

The flow potentials used for the Mohr-Coulomb yield surface and the tension cutoff surface are described below.

Plastic flow on the Mohr-Coulomb yield surface

The flow potential, G, for the Mohr-Coulomb yield surface is chosen as a hyperbolic function in the meridional stress plane and the smooth elliptic function proposed by Menétrey and Willam (1995) in the deviatoric stress plane:

$$G = \sqrt{(\epsilon c|_0 \tan\psi)^2 + (R_{mw}q)^2} - p\tan\psi,$$

where

$$R_{mw}(\Theta, e) = \frac{4(1-e^2)\cos^2\Theta + (2e-1)^2}{2(1-e^2)\cos\Theta + (2e-1)\sqrt{4(1-e^2)\cos^2\Theta + 5e^2 - 4e}} R_{mc}(\frac{\pi}{3}, \phi),$$

and

$$R_{mc}(\frac{\pi}{3}, \phi) = \frac{3 - \sin\phi}{6\cos\phi},$$

ψ is the dilation angle measured in the p–$R_{mw}q$ plane at high confining pressure and can depend on temperature and predefined field variables;

$c|_0$ is the initial cohesion yield stress, $c|_0 = c|_{\bar{\varepsilon}^{pl}=0}$;

Θ is the deviatoric polar angle defined previously;

ϵ is a parameter, referred to as the meridional eccentricity, that defines the rate at which the hyperbolic function approaches the asymptote (the flow potential tends to a straight line in the meridional stress plane as the meridional eccentricity tends to zero); and

e is a parameter, referred to as the deviatoric eccentricity, that describes the "out-of-roundedness" of the deviatoric section in terms of the ratio between the shear stress along the extension meridian ($\Theta = 0$) and the shear stress along the compression meridian ($\Theta = \frac{\pi}{3}$).

A default value of 0.1 is provided for the meridional eccentricity, ϵ.

By default, the deviatoric eccentricity, ***e***, is calculated as

$$e = \frac{3 - \sin\phi}{3 + \sin\phi},$$

where ϕ is the Mohr-Coulomb friction angle; this calculation corresponds to matching the flow potential to the yield surface in both triaxial tension and compression in the deviatoric plane. Alternatively, Abaqus allows you to consider this deviatoric eccentricity as an independent material parameter; in this case you provide its value directly. Convexity and smoothness of the elliptic function requires that $1/2 < e \leq 1$. The upper limit, $e = 1$ (or $\phi = 0°$ when you do not specify the value of ***e***), leads to $R_{mw}(\Theta, e = 1) = R_{mc}(\frac{\pi}{3}, \phi)$, which describes the Mises circle in the deviatoric plane. The lower limit, $e = 1/2$ (or $\phi = 90°$ when you do not specify the value of ***e***), leads to $R_{mw}(\Theta, e = 1/2) = 2R_{mc}(\frac{\pi}{3}, \phi)\cos\Theta$ and would describe the Rankine triangle in the deviatoric plane (this limiting case is not permitted within the Mohr-Coulomb model described here).

This flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. A family of hyperbolic potentials in the meridional stress plane is shown in Figure 23.3.3–3, and the flow potential in the deviatoric stress plane is shown in Figure 23.3.3–4.

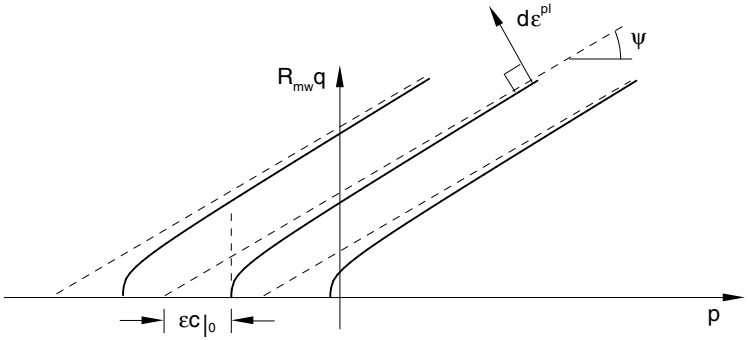


Figure 23.3.3–3 Family of hyperbolic flow potentials in the meridional stress plane.

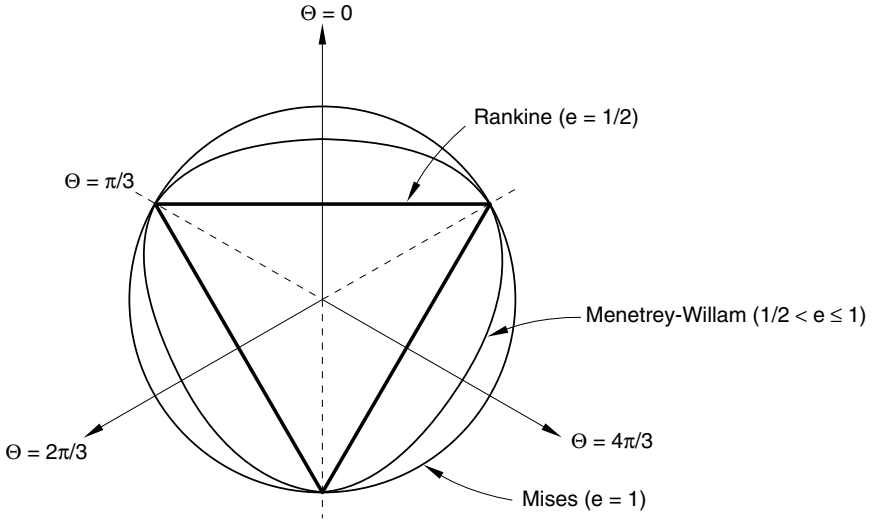


Figure 23.3.3–4 Menétrey-Willam flow potential in the deviatoric stress plane.

Flow in the meridional stress plane can be close to associated when the angle of friction, ϕ, and the angle of dilation, ψ, are equal and the meridional eccentricity, ϵ, is very small; however, flow in this plane is, in general, nonassociated. Flow in the deviatoric stress plane is always nonassociated.

Input File Usage: Use the following option to allow Abaqus to calculate the value of e (default):

∗MOHR COULOMB

Use the following option to specify the value of e directly:

∗MOHR COULOMB, DEVIATORIC ECCENTRICITY=e

Abaqus/CAE Usage: Use the following option to allow Abaqus to calculate the value of e (default):

Property module: material editor: **Mechanical→Plasticity→Mohr Coulomb Plasticity**: **Plasticity**: **Deviatoric eccentricity: Calculated default**

Use the following option to specify the value of e directly:

Property module: material editor: **Mechanical→Plasticity→Mohr Coulomb Plasticity**: **Plasticity**: **Deviatoric eccentricity: Specify:** e

Plastic flow on the Rankine surface

A flow potential that results in a nearly associative flow is chosen for the Rankine surface and is constructed by modifying the Menétrey-Willam potential described earlier:

$$G_t = \sqrt{\left(\epsilon_t \sigma_t|_0\right)^2 + \left(R_t q\right)^2} - p,$$

where

$$R_t(\Theta, e_t) = \frac{1}{3} \frac{4(1-e_t^2)\cos^2\Theta + (2e_t-1)^2}{2(1-e_t^2)\cos\Theta + (2e_t-1)\sqrt{4(1-e_t^2)\cos^2\Theta + 5e_t^2 - 4e_t}},$$

$\sigma_t|_0$ is the initial value of tension cutoff;

ϵ_t is the meridional eccentricity, similar to ϵ defined earlier; and

e_t is the deviatoric eccentricity, similar to e defined earlier.

Abaqus uses values of 0.1 and 0.6, for ϵ_t and e_t, respectively.

Nonassociated flow

Since the plastic flow is nonassociated in general, the use of this Mohr-Coulomb model generally requires the unsymmetric matrix storage and solution scheme in Abaqus/Standard (see "Defining an analysis," Section 6.1.2).

Elements

The Mohr-Coulomb plasticity model can be used with any stress/displacement elements other than one-dimensional elements (beam, pipe, and truss elements) or elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables are available for the Mohr-Coulomb plasticity model:

PEEQ	Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \int \frac{1}{c}\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}^{pl}$, where c is the cohesion yield stress.
PEEQT	Tensile equivalent plastic strain, $\bar{\varepsilon_t}^{pl}$, on the tension cutoff yield surface.

Additional reference

- Menétrey, Ph., and K. J. Willam, "Triaxial Failure Criterion for Concrete and its Generalization," ACI Structural Journal, vol. 92, pp. 311–318, May/June 1995.

23.3.4 CRITICAL STATE (CLAY) PLASTICITY MODEL

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *CLAY PLASTICITY
- *CLAY HARDENING
- "Defining clay plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual
- "Critical state models," Section 4.4.3 of the Abaqus Theory Manual

Overview

The clay plasticity model provided in Abaqus:

- describes the inelastic behavior of the material by a yield function that depends on the three stress invariants, an associated flow assumption to define the plastic strain rate, and a strain hardening theory that changes the size of the yield surface according to the inelastic volumetric strain;
- requires that the elastic part of the deformation be defined by using the linear elastic material model ("Linear elastic behavior," Section 22.2.1) or, in Abaqus/Standard, the porous elastic material model ("Elastic behavior of porous materials," Section 22.3.1) within the same material definition; and
- allows for the hardening law to be defined by a piecewise linear form or, in Abaqus/Standard, by an exponential form.

Yield surface

The model is based on the yield surface

$$\frac{1}{\beta^2}\left(\frac{p}{a}-1\right)^2+\left(\frac{t}{Ma}\right)^2-1=0,$$

where

$p=-\frac{1}{3}\mathrm{trace}\,\boldsymbol{\sigma}$	is the equivalent pressure stress;
$t=\frac{1}{2}q\left[1+\frac{1}{K}-\left(1-\frac{1}{K}\right)\left(\frac{r}{q}\right)^3\right]$	is a deviatoric stress measure;
$q=\sqrt{\frac{3}{2}\mathbf{S}:\mathbf{S}}$	is the Mises equivalent stress;
$r=(\frac{9}{2}\,\mathbf{S}:\mathbf{S}\cdot\mathbf{S})^{\frac{1}{3}}$	is the third stress invariant;
M	is a constant that defines the slope of the critical state line;

β is a constant that is equal to 1.0 on the "dry" side of the critical state line ($t > Mp$) but may be different from 1.0 on the "wet" side of the critical state line ($\beta \neq 1.0$ introduces a different ellipse on the wet side of the critical state line; i.e., a tighter "cap" is obtained if $\beta < 1.0$ as shown in Figure 23.3.4–1);

a is the size of the yield surface (Figure 23.3.4–1); and

K is the ratio of the flow stress in triaxial tension to the flow stress in triaxial compression and determines the shape of the yield surface in the plane of principal deviatoric stresses (the "Π-plane": see Figure 23.3.4–2); Abaqus requires that $0.778 \leq K \leq 1.0$ to ensure that the yield surface remains convex.

The user-defined parameters M, β, and K can depend on temperature θ as well as other predefined field variables, f_i. The model is described in detail in "Critical state models," Section 4.4.3 of the Abaqus Theory Manual.

Input File Usage: *CLAY PLASTICITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Clay Plasticity**

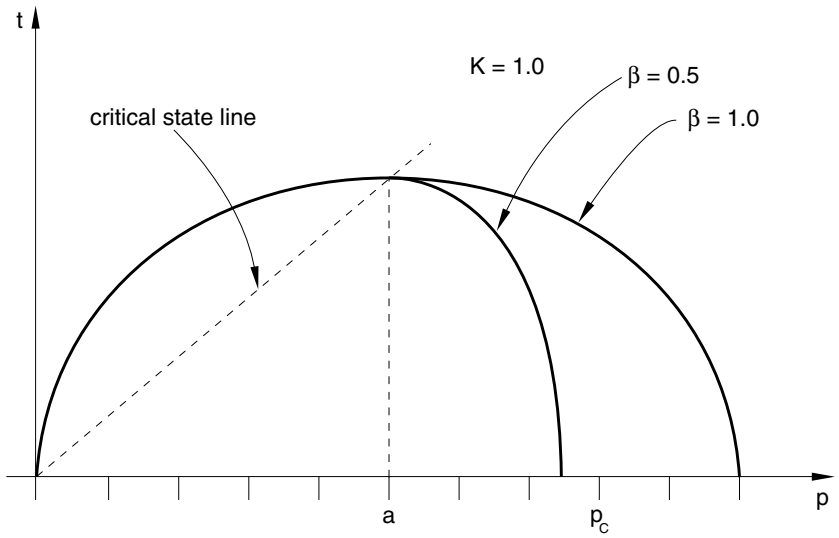


Figure 23.3.4–1 Clay yield surfaces in the p–t plane.

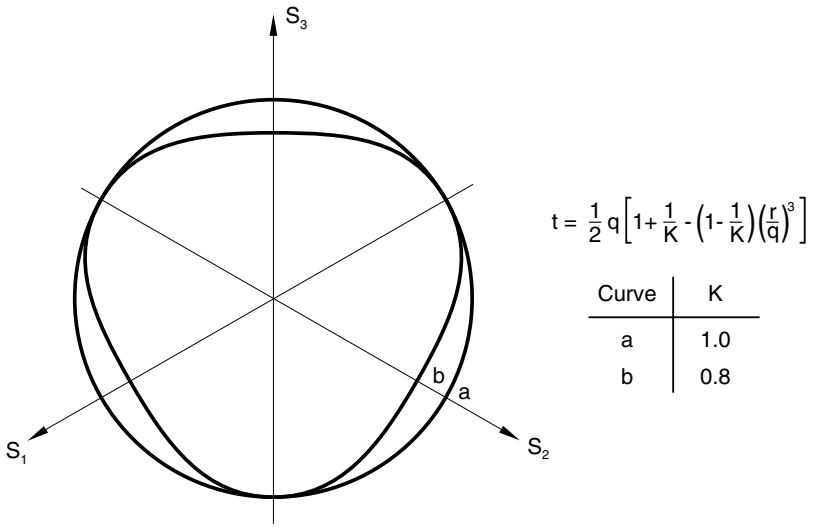


Figure 23.3.4–2 Clay yield surface sections in the Π-plane.

Hardening law

The hardening law can have an exponential form (Abaqus/Standard only), or a piecewise linear form.

Exponential form in Abaqus/Standard

The exponential form of the hardening law is written in terms of some of the porous elasticity parameters and, therefore, can be used only in conjunction with the Abaqus/Standard porous elastic material model. The size of the yield surface at any time is determined by the initial value of the hardening parameter, a_0, and the amount of inelastic volume change that occurs according to the equation

$$a = a_0 \exp\left[(1 + e_0)\frac{1 - J^{pl}}{\lambda - \kappa J^{pl}}\right],$$

where

- J^{pl} is the inelastic volume change (that part of J, the ratio of current volume to initial volume, attributable to inelastic deformation);
- $\kappa(\theta, f_i)$ is the logarithmic bulk modulus of the material defined for the porous elastic material behavior;
- $\lambda(\theta, f_i)$ is the logarithmic hardening constant defined for the clay plasticity material behavior; and
- e_0 is the user-defined initial void ratio ("Defining initial void ratios in a porous medium" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Specifying the initial size of the yield surface directly

The initial size of the yield surface is defined for clay plasticity by specifying the hardening parameter, a_0, as a tabular function or by defining it analytically.

a_0 can be defined along with λ, $\boldsymbol{M}$, β, and $\boldsymbol{K}$, as a tabular function of temperature and other predefined field variables. However, a_0 is a function only of the initial conditions; it will not change if temperatures and field variables change during the analysis.

Input File Usage: Use all of the following options:

∗INITIAL CONDITIONS, TYPE=RATIO
∗POROUS ELASTIC
∗CLAY PLASTICITY, HARDENING=EXPONENTIAL

Abaqus/CAE Usage: Use all of the following options:

Property module: material editor:
Mechanical→Elasticity→Porous Elastic
Mechanical→Plasticity→Clay Plasticity: **Hardening: Exponential**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Void ratio** for the **Types for Selected Step**

Specifying the initial size of the yield surface indirectly

The hardening parameter a_0 can be defined indirectly by specifying e_1, which is the intercept of the virgin consolidation line with the void ratio axis in the plot of void ratio, $\boldsymbol{e}$, versus the logarithm of the effective pressure stress, $\ln p$ (Figure 23.3.4–3). If this method is used, a_0 is defined by

$$a_0 = \frac{1}{2} \exp\left(\frac{e_1 - e_0 - \kappa \ln p_0}{\lambda - \kappa} \right),$$

where p_0 is the user-defined initial value of the equivalent hydrostatic pressure stress (see "Defining initial stresses" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). You define e_1, λ, $\boldsymbol{M}$, β, and $\boldsymbol{K}$; all the parameters except e_1 can be dependent on temperature and other predefined field variables. However, a_0 is a function only of the initial conditions; it will not change if temperatures and field variables change during the analysis.

Input File Usage: Use all of the following options:

∗INITIAL CONDITIONS, TYPE=RATIO
∗INITIAL CONDITIONS, TYPE=STRESS
∗POROUS ELASTIC
∗CLAY PLASTICITY, HARDENING=EXPONENTIAL, INTERCEPT=e_1

Abaqus/CAE Usage: Use all of the following options:

Property module: material editor:
Mechanical→Elasticity→Porous Elastic

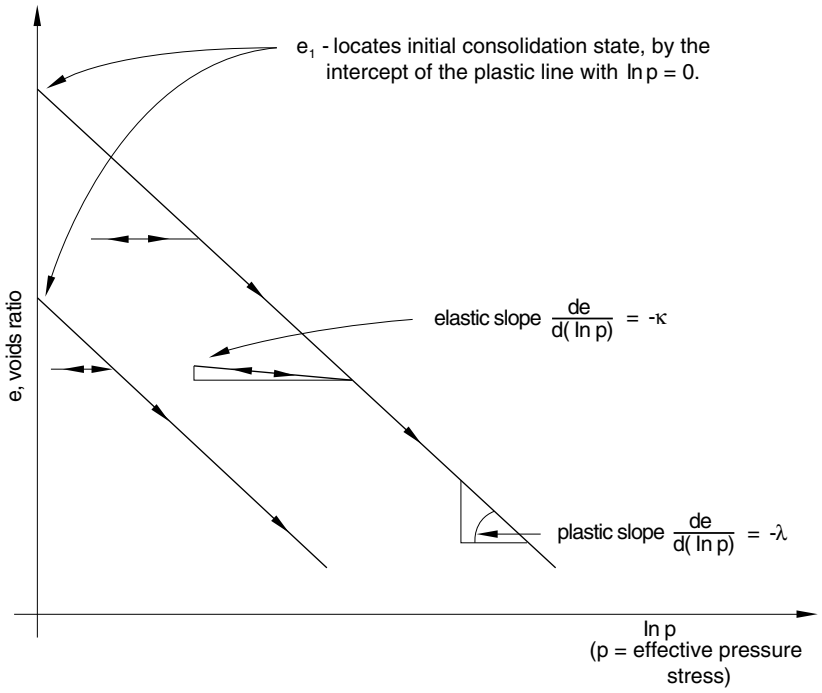

e1 - locates initial consolidation state, by the intercept of the plastic line with ln p = 0.
elastic slope de / d(ln p) = -κ
plastic slope de / d(ln p) = -λ
e, voids ratio
ln p
(p = effective pressure stress)

Figure 23.3.4–3 Pure compression behavior for clay model.

Mechanical→Plasticity→Clay Plasticity: **Hardening: Exponential**, **Intercept:** e_1

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Void ratio** for the **Types for Selected Step**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Stress** for the **Types for Selected Step**

Piecewise linear form

If the piecewise linear form of the hardening rule is used, the user-defined relationship relates the yield stress in hydrostatic compression, p_c, to the corresponding volumetric plastic strain, $\varepsilon_{\text{vol}}^{pl}$ (Figure 23.3.4–4):

$$p_c = p_c(\varepsilon_{\text{vol}}^{pl}).$$

The evolution parameter, a, is then given by

$$a = \frac{p_c}{(1+\beta)}.$$

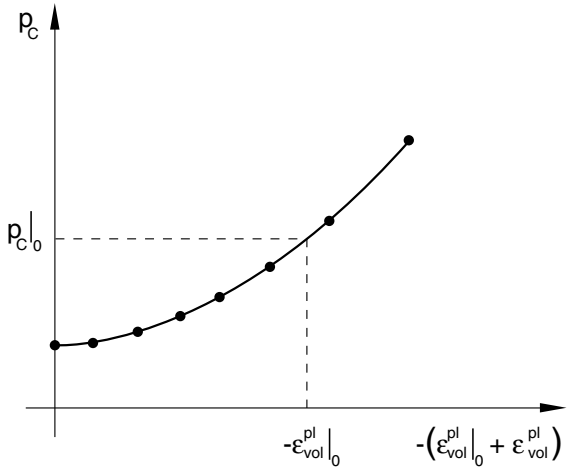


Figure 23.3.4–4 Typical piecewise linear clay hardening/softening curve.

The volumetric plastic strain axis has an arbitrary origin: $\varepsilon^{pl}_{\text{vol}}|_0$ is the position on this axis corresponding to the initial state of the material, thus defining the initial hydrostatic pressure, $p_c|_0$, and, hence, the initial yield surface size, a_0. This relationship is defined in tabular form as clay hardening data. The range of values for which p_c is defined should be sufficient to include all values of equivalent pressure stress to which the material will be subjected during the analysis.

This form of the hardening law can be used in conjunction with either the linear elastic or, in Abaqus/Standard, the porous elastic material models. This is the only form of the hardening law supported in Abaqus/Explicit

Input File Usage: Use both of the following options:

*CLAY PLASTICITY, HARDENING=TABULAR
*CLAY HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Clay Plasticity**: **Hardening: Tabular**, **Suboptions→Clay Hardening**

Calibration

At least two experiments are required to calibrate the simplest version of the Cam-clay model: a hydrostatic compression test (an oedometer test is also acceptable) and a triaxial compression test (more than one triaxial test is useful for a more accurate calibration).

Hydrostatic compression tests

The hydrostatic compression test is performed by pressurizing the sample equally in all directions. The applied pressure and the volume change are recorded.

The onset of yielding in the hydrostatic compression test immediately provides the initial position of the yield surface, a_0. The logarithmic bulk moduli, κ and λ, are determined from the hydrostatic compression experimental data by plotting the logarithm of pressure versus void ratio. The void ratio, e, is related to the measured volume change as

$$J = \exp(\varepsilon_{vol}) = \frac{1+e}{1+e_0}.$$

The slope of the line obtained for the elastic regime is $-\kappa$, and the slope in the inelastic range is $-\lambda$. For a valid model $\lambda > \kappa$.

Triaxial tests

Triaxial compression experiments are performed using a standard triaxial machine where a fixed confining pressure is maintained while the differential stress is applied. Several tests covering the range of confining pressures of interest are usually performed. Again, the stress and strain in the direction of loading are recorded, together with the lateral strain so that the correct volume changes can be calibrated.

The triaxial compression tests allow the calibration of the yield parameters M and β. M is the ratio of the shear stress, q, to the pressure stress, p, at critical state and can be obtained from the stress values when the material has become perfectly plastic (critical state). β represents the curvature of the cap part of the yield surface and can be calibrated from a number of triaxial tests at high confining pressures (on the "wet" side of critical state). β must be between 0.0 and 1.0.

To calibrate the parameter K, which controls the yield dependence on the third stress invariant, experimental results obtained from a true triaxial (cubical) test are necessary. These results are generally not available, and you may have to guess (the value of K is generally between 0.8 and 1.0) or ignore this effect.

Unloading measurements

Unloading measurements in hydrostatic and triaxial compression tests are useful to calibrate the elasticity, particularly in cases where the initial elastic region is not well defined. From these we can identify whether a constant shear modulus or a constant Poisson's ratio should be used and what their values are.

Initial conditions

If an initial stress at a point is given (see "Defining initial stresses" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1) such that the stress point lies outside the initially defined yield surface, Abaqus will try to adjust the initial position of the surface to make the stress point lie on it and issue a warning. However, if the stress point is such that the equivalent pressure stress, p, is negative, an error message will be issued and execution will be terminated.

Elements

The clay plasticity model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus (“Abaqus/Standard output variable identifiers,” Section 4.2.1, and “Abaqus/Explicit output variable identifiers,” Section 4.2.2), the following variable has special meaning for material points in the clay plasticity model:

PEEQ	Center of the yield surface, $\boldsymbol{a}$.

23.3.5 CRUSHABLE FOAM PLASTICITY MODELS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- "Rate-dependent yield," Section 23.2.3
- *CRUSHABLE FOAM
- *CRUSHABLE FOAM HARDENING
- *RATE DEPENDENT
- "Defining crushable foam plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The crushable foam plasticity models:

- are intended for the analysis of crushable foams that are typically used as energy absorption structures;
- can be used to model crushable materials other than foams (such as balsa wood);
- are used to model the enhanced ability of a foam material to deform in compression due to cell wall buckling processes (it is assumed that the resulting deformation is not recoverable instantaneously and can, thus, be idealized as being plastic for short duration events);
- can be used to model the difference between a foam material's compressive strength and its much smaller tensile bearing capacity resulting from cell wall breakage in tension;
- must be used in conjunction with the linear elastic material model ("Linear elastic behavior," Section 22.2.1);
- can be used when rate-dependent effects are important; and
- are intended to simulate material response under essentially monotonic loading.

Elastic and plastic behavior

The elastic part of the response is specified as described in "Linear elastic behavior," Section 22.2.1. Only linear isotropic elasticity can be used.

For the plastic part of the behavior, the yield surface is a Mises circle in the deviatoric stress plane and an ellipse in the meridional (p–q) stress plane. Two hardening models are available: the volumetric hardening model, where the point on the yield ellipse in the meridional plane that represents hydrostatic tension loading is fixed and the evolution of the yield surface is driven by the volumetric compacting plastic strain, and the isotropic hardening model, where the yield ellipse is centered at the origin in the

p*–*q stress plane and evolves in a geometrically self-similar manner. This phenomenological isotropic model was originally developed for metallic foams by Deshpande and Fleck (2000).

The hardening curve must describe the uniaxial compression yield stress as a function of the corresponding plastic strain. In defining this dependence at finite strains, "true" (Cauchy) stress and logarithmic strain values should be given. Both models predict similar behavior for compression-dominated loading. However, for hydrostatic tension loading the volumetric hardening model assumes a perfectly plastic behavior, while the isotropic hardening model predicts the same behavior in both hydrostatic tension and hydrostatic compression.

Crushable foam model with volumetric hardening

The crushable foam model with volumetric hardening uses a yield surface with an elliptical dependence of deviatoric stress on pressure stress. It assumes that the evolution of the yield surface is controlled by the volumetric compacting plastic strain experienced by the material.

Yield surface

The yield surface for the volumetric hardening model is defined as

$$F = \sqrt{q^2 + \alpha^2 (p - p_0)^2} - B = 0,$$

where

$p = -\frac{1}{3}\text{trace}\,\boldsymbol{\sigma}$	is the pressure stress,
$q = \sqrt{\frac{3}{2}\,\mathbf{S} : \mathbf{S}}$	is the Mises stress,
$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}$	is the deviatoric stress,
A	is the size of the (horizontal) ***p***-axis of the yield ellipse,
$B = \alpha\, A = \alpha\, \frac{p_c + p_t}{2}$	is the size of the (vertical) ***q***-axis of the yield ellipse,
$\alpha = B/A$	is the shape factor of the yield ellipse that defines the relative magnitude of the axes,
$p_0 = \frac{p_c - p_t}{2}$	is the center of the yield ellipse on the ***p***-axis,
p_t	is the strength of the material in hydrostatic tension, and
p_c	is the yield stress in hydrostatic compression (p_c is always positive).

The yield surface represents the Mises circle in the deviatoric stress plane and is an ellipse on the meridional stress plane, as depicted in Figure 23.3.5–1.

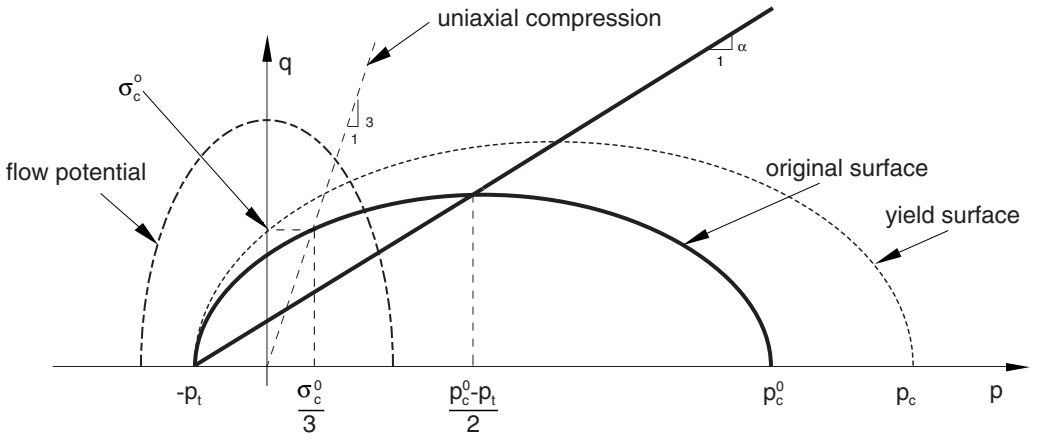


Figure 23.3.5–1 Crushable foam model with volumetric hardening: yield surface and flow potential in the ***p*–*q*** stress plane.

The yield surface evolves in a self-similar fashion (constant α); and the shape factor can be computed using the initial yield stress in uniaxial compression, σ_c^0, the initial yield stress in hydrostatic compression, p_c^0 (the initial value of p_c), and the yield strength in hydrostatic tension, p_t:

$$\alpha = \frac{3k}{\sqrt{(3k_t + k)(3 - k)}} \qquad \text{with} \qquad k = \frac{\sigma_c^0}{p_c^0} \quad \text{and} \quad k_t = \frac{p_t}{p_c^0}.$$

For a valid yield surface the choice of strength ratios must be such that $0 < k < 3$ and $k_t \geq 0$. If this is not the case, Abaqus will issue an error message and terminate execution.

To define the shape of the yield surface, you provide the values of k and k_t. If desired, these variables can be defined as a tabular function of temperature and other predefined field variables.

Input File Usage: *CRUSHABLE FOAM, HARDENING=VOLUMETRIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Hardening: Volumetric**

Calibration

To use this model, one needs to know the initial yield stress in uniaxial compression, σ_c^0; the initial yield stress in hydrostatic compression, p_c^0; and the yield strength in hydrostatic tension, p_t. Since foam materials are rarely tested in tension, it is usually necessary to guess the magnitude of the strength of the foam in hydrostatic tension, p_t. The choice of tensile strength should not have a strong effect on the numerical results unless the foam is stressed in hydrostatic tension. A common approximation is to set p_t equal to 5% to 10% of the initial yield stress in hydrostatic compression p_c^0; thus, k_t= 0.05 to 0.10.

Flow potential

The plastic strain rate for the volumetric hardening model is assumed to be

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\bar{\varepsilon}}^{pl}\,\frac{\partial G}{\partial \boldsymbol{\sigma}},$$

where G is the flow potential, chosen in this model as

$$G = \sqrt{q^2 + \frac{9}{2}\,p^2},$$

and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate defined as

$$\dot{\bar{\varepsilon}}^{pl} = \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}}{G}.$$

The equivalent plastic strain rate is related to the rate of axial plastic strain, $\dot{\varepsilon}^{pl}_{\text{axial}}$, in uniaxial compression by

$$\dot{\bar{\varepsilon}}^{pl} = \sqrt{\frac{2}{3}}\,\dot{\varepsilon}^{pl}_{\text{axial}}.$$

A geometrical representation of the flow potential in the p–q stress plane is shown in Figure 23.3.5–1. This potential gives a direction of flow that is identical to the stress direction for radial paths. This is motivated by simple laboratory experiments that suggest that loading in any principal direction causes insignificant deformation in the other directions. As a result, the plastic flow is nonassociative for the volumetric hardening model. For more details regarding plastic flow, see "Plasticity models: general discussion," Section 4.2.1 of the Abaqus Theory Manual.

Nonassociated flow

The nonassociated plastic flow rule makes the material stiffness matrix unsymmetric; therefore, the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see "Defining an analysis," Section 6.1.2). Usage of this scheme is especially important when large regions of the model are expected to flow plastically.

Hardening

The yield surface intersects the p-axis at $-p_t$ and p_c. We assume that p_t remains fixed throughout any plastic deformation process. By contrast, the compressive strength, p_c, evolves as a result of compaction (increase in density) or dilation (reduction in density) of the material. The evolution of the yield surface can be expressed through the evolution of the yield surface size on the hydrostatic stress axis, $p_c + p_t$, as a function of the value of volumetric compacting plastic strain, $-\varepsilon^{pl}_{\text{vol}}$. With p_t constant, this relation can be obtained from user-provided uniaxial compression test data using

$$p_c(\varepsilon^{pl}_{\text{vol}}) = \frac{\sigma_c(\varepsilon^{pl}_{\text{axial}}) \left[\sigma_c(\varepsilon^{pt}_{\text{axial}}) \left(\frac{1}{\alpha^2} + \frac{1}{9} \right) + \frac{p_t}{3} \right]}{p_t + \frac{\sigma_c(\varepsilon^{pl}_{\text{axial}})}{3}}$$

along with the fact that $\varepsilon^{pl}_{\text{axial}} = \varepsilon^{pl}_{\text{vol}}$ in uniaxial compression for the volumetric hardening model. Thus, you provide input to the hardening law by specifying, in the usual tabular form, only the value of the yield stress in uniaxial compression as a function of the absolute value of the axial plastic strain. The table must start with a zero plastic strain (corresponding to the virgin state of the material), and the tabular entries must be given in ascending magnitude of $\varepsilon^{pl}_{\text{axial}}$. If desired, the yield stress can also be a function of temperature and other predefined field variables.

Input File Usage: *CRUSHABLE FOAM HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Suboptions→Foam Hardening**

Rate dependence

As strain rates increase, many materials show an increase in the yield stress. For many crushable foam materials this increase in yield stress becomes important when the strain rates are in the range of 0.1–1 per second and can be very important if the strain rates are in the range of 10–100 per second, as commonly occurs in high-energy dynamic events.

Two methods for specifying strain-rate-dependent material behavior are available in Abaqus as discussed below. Both methods assume that the shapes of the hardening curves at different strain rates are similar, and either can be used in static or dynamic procedures. When rate dependence is included, the static stress-strain hardening behavior must be specified for the crushable foam as described above.

Overstress power law

You can specify a Cowper-Symonds overstress power law that defines strain rate dependence. This law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D\left(R - 1 \right)^n \qquad \text{for} \qquad R \geq 1,$$

with

$$R \equiv \frac{\bar{B}}{B},$$

where B is the size of the static yield surface and $\bar{B}$ is the size of the yield surface at a nonzero strain rate. The ratio R can be written as

$$R-1=\left(r-1\right)\frac{3k_t+r\left[k+k_t(3-k)\right]}{(1+k_t)(3k_t+r\,k)},$$

where r is the uniaxial compression yield stress ratio defined by

$$r\equiv\frac{\bar{\sigma}_c}{\sigma_c}.$$

σ_c, specified as part of the crushable foam hardening definition, is the uniaxial compression yield stress at a given value of $\varepsilon^{pl}_{\text{axial}}$ for the experiment with the lowest strain rate and can depend on temperature and predefined field variables; D and n are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: Use both of the following options:

*CRUSHABLE FOAM HARDENING
*RATE DEPENDENT, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Suboptions→Foam Hardening**; **Suboptions→Rate Dependent**: **Hardening: Power Law**

The power-law rate dependency can be rewritten in the following form

$$\ln\left(R-1\right)=\frac{1}{n}\,\ln\dot{\bar{\varepsilon}}^{pl}-\frac{1}{n}\,\ln D.$$

The procedure outlined below can be followed to obtain the material parameters D and n based on the uniaxial compression test data.

1. Compute R using the uniaxial compression yield stress ratio, r.
2. Convert the rate of the axial plastic strain, $\dot{\varepsilon}^{pl}_{\text{axial}}$, to the corresponding equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$.
3. Plot $\ln(R-1)$ versus $\ln(\dot{\bar{\varepsilon}}^{pl})$. If the curve can be approximated by a straight line such as that shown in Figure 23.3.5–2, the overstress power law is suitable. The slope of the line is $1/n$, and the intercept of the $\ln(R-1)$ axis is $-(1/n)\ln D$.

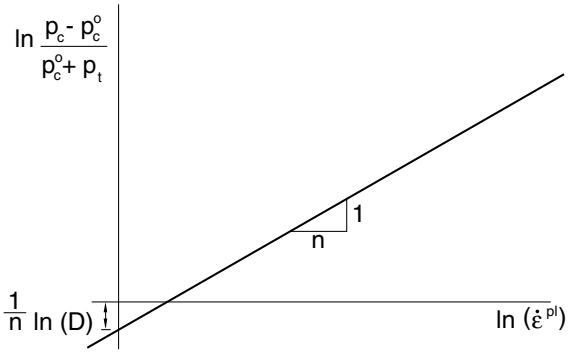


Figure 23.3.5–2 Calibration of overstress power law data.

Tabular input of yield ratio

Rate-dependent behavior can alternatively be specified by giving a table of the ratio $r = \bar{\sigma}_c/\sigma_c$ as a function of the absolute value of the rate of the axial plastic strain and, optionally, temperature and predefined field variables.

Input File Usage: Use both of the following options:

*CRUSHABLE FOAM HARDENING
*RATE DEPENDENT, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Suboptions→Foam Hardening**; **Suboptions→Rate Dependent**: **Hardening: Yield Ratio**

Initial conditions

When we need to study the behavior of a material that has already been subjected to some hardening, Abaqus allows you to prescribe initial conditions for the volumetric compacting plastic strain, $-\varepsilon_{\text{vol}}^{pl}$ (see "Defining initial values of state variables for plastic hardening" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Crushable foam model with isotropic hardening

The isotropic hardening model uses a yield surface that is an ellipse centered at the origin in the p–q stress plane. The yield surface evolves in a self-similar manner, and the evolution is governed by the equivalent plastic strain (to be defined later).

Yield surface

The yield surface for the isotropic hardening model is defined as

$$F = \sqrt{q^2 + \alpha^2\, p^2} - B = 0,$$

where

$p = -\frac{1}{3}\text{trace}\,\boldsymbol{\sigma}$	is the pressure stress,
$q = \sqrt{\frac{3}{2}\,\mathbf{S} : \mathbf{S}}$	is the Mises stress,
$\mathbf{S} = \boldsymbol{\sigma} + p\,\mathbf{I}$	is the deviatoric stress,
$B = \alpha\, p_c = \sigma_c \sqrt{1 + \left(\frac{\alpha}{3}\right)^2}$	is the size of the (vertical) ***q***-axis of the yield ellipse,
α	is the shape factor of the yield ellipse that defines the relative magnitude of the axes,
p_c	is the yield stress in hydrostatic compression, and
σ_c	is the absolute value of the yield stress in uniaxial compression.

The yield surface represents the Mises circle in the deviatoric stress plane. The shape of the yield surface in the meridional stress plane is depicted in Figure 23.3.5–3. The shape factor, α, can be computed using the initial yield stress in uniaxial compression, σ_c^0, and the initial yield stress in hydrostatic compression, p_c^0 (the initial value of p_c), using the relation:

$$\alpha = \frac{3k}{\sqrt{9 - k^2}} \qquad \text{with} \qquad k = \frac{\sigma_c^0}{p_c^0}.$$

To define the shape of the yield ellipse, you provide the value of ***k***. For a valid yield surface the strength ratio must be such that $0 \le k < 3$. The particular case of $k = 0$ corresponds to the Mises plasticity. In general, the initial yield strengths in uniaxial compression and in hydrostatic compression, σ_c^0 and p_c^0, can be used to calculate the value of ***k***. However, in many practical cases the stress versus strain response curves of crushable foam materials do not show clear yielding points, and the initial yield stress values cannot be determined exactly. Many of these response curves have a horizontal plateau—the yield stress is nearly a constant for a significantly large range of plastic strain values. If you have data from both uniaxial compression and hydrostatic compression, the plateau values of the two experimental curves can be used to calculate the ratio of ***k***.

Input File Usage: *CRUSHABLE FOAM, HARDENING=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Hardening: Isotropic**

Flow potential

The flow potential for the isotropic hardening model is chosen as

$$G = \sqrt{q^2 + \beta^2\, p^2},$$

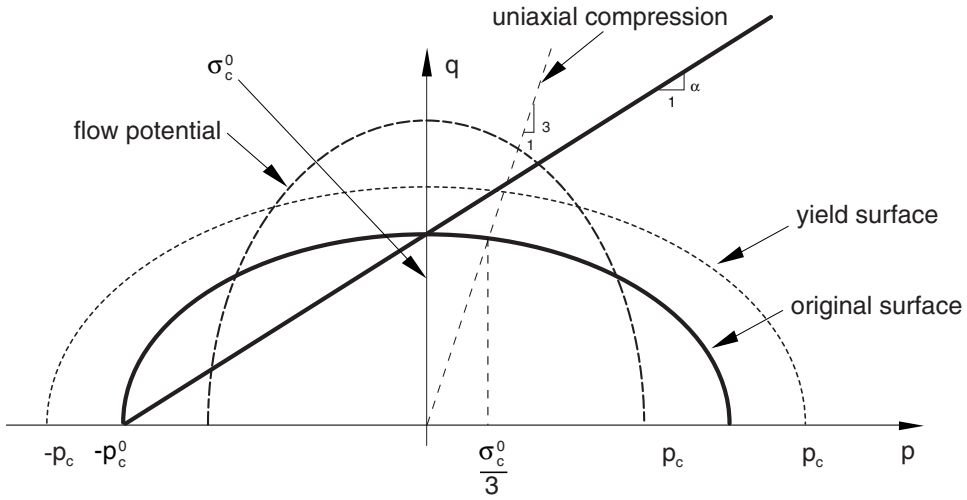


Figure 23.3.5–3 Crushable foam model with isotropic hardening: yield surface and flow potential in the ***p*–*q*** stress plane.

where β represents the shape of the flow potential ellipse on the ***p*–*q*** stress plane. It is related to the plastic Poisson's ratio, ν_p, via

$$\beta = \frac{3}{\sqrt{2}} \sqrt{\frac{1 - 2\nu_p}{1 + \nu_p}}.$$

The plastic Poisson's ratio, which is the ratio of the transverse to the longitudinal plastic strain under uniaxial compression, must be in the range of -1 and 0.5; and the upper limit ($\nu_p = 0.5$) corresponds to the case of incompressible plastic flow ($\beta = 0$). For many low-density foams the plastic Poisson's ratio is nearly zero, which corresponds to a value of $\beta \approx 2.12$.

The plastic flow is associated when the value of α is the same as that of β. By default, the plastic flow is nonassociated to allow for the independent calibrations of the shape of the yield surface and the plastic Poisson's ratio. If you have information only about the plastic Poisson's ratio and choose to use associated plastic flow, the yield stress ratio ***k*** can be calculated from

$$k = \sqrt{3(1 - 2\,\nu_p)}.$$

Alternatively, if only the shape of the yield surface is known and you choose to use associated plastic flow, the plastic Poisson's ratio can be obtained from

$$\nu_p = \frac{3 - k^2}{6}.$$

You provide the value of ν_p.

Input File Usage: *CRUSHABLE FOAM, HARDENING=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Hardening: Isotropic**

Hardening

A simple uniaxial compression test is sufficient to define the evolution of the yield surface. The hardening law defines the value of the yield stress in uniaxial compression as a function of the absolute value of the axial plastic strain. The piecewise linear relationship is entered in tabular form. The table must start with a zero plastic strain (corresponding to the virgin state of the materials), and the tabular entries must be given in ascending magnitude of $\varepsilon^{pl}_{\text{axial}}$. For values of plastic strain greater than the last user-specified value, the stress-strain relationship is extrapolated based on the last slope computed from the data. If desired, the yield stress can also be a function of temperature and other predefined field variables.

Input File Usage: *CRUSHABLE FOAM HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Suboptions→Foam Hardening**

Rate dependence

As strain rates increase, many materials show an increase in the yield stress. For many crushable foam materials this increase in yield stress becomes important when the strain rates are in the range of 0.1–1 per second and can be very important if the strain rates are in the range of 10–100 per second, as commonly occurs in high-energy dynamic events.

Two methods for specifying strain-rate-dependent material behavior are available in Abaqus as discussed below. Both methods assume that the shapes of the hardening curves at different strain rates are similar, and either can be used in static or dynamic procedures. When rate dependence is included, the static stress-strain hardening behavior must be specified for the crushable foam as described above.

Overstress power law

You can specify a Cowper-Symonds overstress power law that defines strain rate dependence. This law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D\left(R-1\right)^n \qquad \text{for} \qquad R \geq 1,$$

with

$$R \equiv \frac{\bar{\sigma}_c}{\sigma_c},$$

where σ_c, specified as part of the crushable foam hardening definition, is the static uniaxial compression yield stress at a given value of $\varepsilon^{pl}_{\text{axial}}$ for the experiment with the lowest strain rate, and $\bar{\sigma}_c$ is the yield

stress at a nonzero strain rate. $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate, which is equal to the rate of the axial plastic strain in uniaxial compression for the isotropic hardening model.

The power-law rate dependency can be rewritten in the following form

$$\ln\left(R-1\right) = \frac{1}{n}\,\ln\dot{\bar{\varepsilon}}^{pl} - \frac{1}{n}\,\ln D.$$

Plot $\ln(R-1)$ versus $\ln(\dot{\varepsilon}^{pl}_{\text{axial}})$. If the curve can be approximated by a straight line such as that shown in Figure 23.3.5–2, the overstress power law is suitable. The slope of the line is $1/n$, and the intercept of the $\ln(R-1)$ axis is $-(1/n)\ln D$. The material parameters D and n can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: Use both of the following options:

*CRUSHABLE FOAM HARDENING
*RATE DEPENDENT, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Suboptions→Foam Hardening**; **Suboptions→Rate Dependent**: **Hardening: Power Law**

Tabular input of yield ratio

Rate-dependent behavior can alternatively be specified by giving a table of the ratio R as a function of the absolute value of the rate of the axial plastic strain and, optionally, temperature and predefined field variables.

Input File Usage: Use both of the following options:

*CRUSHABLE FOAM HARDENING
*RATE DEPENDENT, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Crushable Foam**: **Suboptions→Foam Hardening**; **Suboptions→Rate Dependent**: **Hardening: Yield Ratio**

Elements

The crushable foam plasticity model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements. This model cannot be used with elements for which the stress state is assumed to be planar (plane stress, shell, and membrane elements) or with beam, pipe, or truss elements.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variable has special meaning for the crushable foam plasticity model:

PEEQ — For the volumetric hardening model, PEEQ is the volumetric compacting plastic strain defined as $-\varepsilon_{\text{vol}}^{pl}$. For the isotropic hardening model, PEEQ is the equivalent plastic strain defined as $\bar{\varepsilon}^{pl} \equiv \int \frac{\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}^{pl}}{\sigma_c}$, where σ_c is the uniaxial compression yield stress.

The volumetric plastic strain, $\varepsilon_{\text{vol}}^{pl}$, is the trace of the plastic strain tensor; you can also calculate it as the sum of direct plastic strain components.

For the volumetric hardening model, the initial values of the volumetric compacting plastic strain can be specified for elements that use the crushable foam material model, as described above. The volumetric compacting plastic strain (output variable PEEQ) provided by Abaqus then contains the initial value of the volumetric compacting plastic strain plus any additional volumetric compacting plastic strain due to plastic straining during the analysis. However, the plastic strain tensor (output variable PE) contains only the amount of straining due to deformation during the analysis.

Additional reference

- Deshpande, V. S., and N. A. Fleck, "Isotropic Constitutive Model for Metallic Foams," Journal of the Mechanics and Physics of Solids, vol. 48, pp. 1253–1276, 2000.

23.4 Fabric materials

- “Fabric material behavior,” Section 23.4.1

23.4.1 FABRIC MATERIAL BEHAVIOR

Product: Abaqus/Explicit

References

- "Material library: overview," Section 21.1.1
- "Elastic behavior: overview," Section 22.1.1
- "VFABRIC," Section 1.2.3 of the Abaqus User Subroutines Reference Manual
- *FABRIC
- *UNIAXIAL
- *LOADING DATA
- *UNLOADING DATA
- *EXPANSION
- *DENSITY
- *INITIAL CONDITIONS

Overview

The fabric material model:

- is anisotropic and nonlinear;
- is a phenomenological model that captures the mechanical response of a woven fabric made of yarns in the fill and the warp directions;
- is valid for materials that exhibit two "structural" directions that may not be orthogonal to each other with deformation;
- defines the local fabric stresses as a function of change in angle between the fibers (shear strain) and the nominal strains along the yarn directions;
- allows for the computation of local fabric stresses based on test data or through user subroutine **VFABRIC**, which can be used to define a complex constitutive model; and
- requires that geometric nonlinearity be accounted for during the analysis step ("General and linear perturbation procedures," Section 6.1.3), since it is intended for finite-strain applications.

The fabric material model defined based on test data:

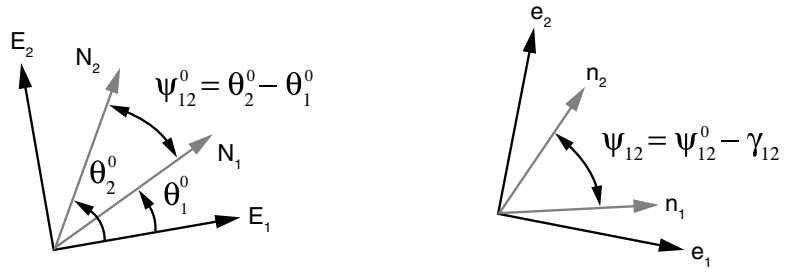
- assumes that the responses along the fill and the warp directions are independent of each other and that the shear response is independent of the direct response along the yarns;
- can include separate loading and unloading responses;
- can exhibit nonlinear elastic behavior, damaged elastic behavior, or elastic-plastic type behavior with permanent deformation upon complete unloading;
- can deform elastically to large tensile and shear strains; and
- can have properties that depend on temperature and/or other field variables.

Fabric material behavior

Woven fabrics are used in a number of engineering applications across various industries, including such products as automobile airbags; flexible structures like boat sails and parachutes; reinforcement in composites; architectural expressions in building roof structures; protective vests for military, police, and other security circles; and protective layers around the fuselage in planes.

Woven fabrics consist of yarns woven in the fill and the warp directions. The yarn is crimped, or curved, as it is woven up and down over the cross yarns. The nonlinear mechanical behavior of the fabric arises from different sources: the nonlinear response of the individual yarns, the exchange of crimp between the fill and the warp yarns as they are stretched, and the contact and friction between the yarns in cross directions and between the yarns in the same direction. In general, the fabric exhibits a significant stiffness only along the yarn directions under tension. The tensile response in the fill and warp directions may be coupled due to the crimp exchange mentioned above. Under in-plane shear deformation, the fill and warp direction yarns rotate with respect to each other. The resistance increases with shear deformation as lateral contact is formed between the yarns in each direction. The fabrics typically have negligible stiffness in bending and in-plane compression.

The behavior of woven fabrics is modeled phenomenologically in Abaqus/Explicit to capture the nonlinear anisotropic behavior of the fabric. The planar kinematic state of a given fabric is described in terms of the nominal direct strains in the fabric plane along the fill and the warp directions and the angle between the two yarn directions. The material orthogonal basis and the yarn local directions are illustrated in Figure 23.4.1–1 showing the reference and the deformed configurations.



(a) Reference configuration

(b) Deformed configuration

Figure 23.4.1–1 Fabric kinematics

The engineering nominal shear strain, γ_{12}, is defined as the change in angle, ψ_{12}, between the two yarn directions going from the reference to the deformed configuration. The nominal strains along the yarn directions n_1 and n_2 in the deformed configuration are computed from the respective yarn stretch values, λ_1 and λ_2. The corresponding nominal stress components T_{11}, T_{22}, and T_{12} are defined as the work conjugate of the above nominal strains. The fabric nominal stress, $\mathbf{T}$, is converted by Abaqus to the Cauchy stress, σ; and the subsequent internal forces arising from the fabric deformation are computed.

You can obtain output of the fabric nominal strains, the fabric nominal stresses, and the regular Cauchy stresses. The relationship between the Cauchy stress, σ, and the nominal stress, $\mathbf{T}$, is

$$J\sigma = \lambda_1 T_{11}\mathbf{n_1}\mathbf{n_1} + \lambda_2 T_{22}\mathbf{n_2}\mathbf{n_2} + T_{12}\csc(\psi_{12})(\mathbf{n_1}\mathbf{n_2} + \mathbf{n_2}\mathbf{n_1}) - T_{12}\cot(\psi_{12})(\mathbf{n_1}\mathbf{n_1} + \mathbf{n_2}\mathbf{n_2}),$$

where J is the volumetric Jacobian.

Either experimental data or a user subroutine, **VFABRIC**, can be used to characterize the Abaqus/Explicit fabric material model, providing the nominal fabric stress as a function of the nominal fabric strains. The user subroutine allows for building a complex material model taking into account both the fabric structural parameters such as yarn spacing, yarn cross-section shape, etc. and the yarn material properties. The test data–based fabric model makes some simplifying assumptions but allows for nonlinear response including energy loss. The two models are presented below in detail. Both models capture the wrinkling of fabric under compression only in a smeared fashion.

The application of fabric material in a crash simulation is illustrated in "Side curtain airbag impactor test," Section 3.3.2 of the Abaqus Example Problems Manual.

Test data–based fabric materials

The fabric material model based on test data assumes that the responses along the fill and the warp directions are independent of each other and that the shear response is independent of the direct response along the yarn. Hence, each component-wise fabric stress response depends only on the fabric strain in that component. Thus, the overall behavior of the fabric consists of three independent component-wise responses: namely, the direct response along the fill yarn to the nominal strain in the fill yarn, the direct response along the warp yarn to the nominal strain in the warp yarn, and the shear response to the change in angle between the two yarns.

Within each component you must provide test data defining the response of the fabric. To fully define the fabric response, the test data must cover all of the following attributes:

- Within a component, separate test data can be defined for the fabric response in the tensile direction and in the compressive direction.
- Within a deformation direction (tension or compression), both loading and unloading test data can be provided.
- The loading and unloading test data can be classified according to three available behavior types: nonlinear elastic behavior, damaged elastic behavior, or elastic-plastic type behavior with permanent deformation. The behavior type determines how the fabric transitions from its loading response to its unloading response.

When elastic, the test data in a particular component can also be rate dependent. When separate loading and unloading paths are required, the test data for the two deformation directions (tension and compression) must be given separately. Otherwise, the data for both tension and compression may be given in a single table.

Input File Usage: Use the following options to define a fabric material using test data:

∗FABRIC
∗UNIAXIAL, COMPONENT=*component*

∗LOADING DATA, DIRECTION=*deformation direction*,
TYPE=*behavior type*
data lines to define loading data
∗UNLOADING DATA
data lines to define unloading data

Repeat all of the options underneath ∗FABRIC as often as necessary to account for each component and deformation direction.

Specifying uniaxial behavior in a component direction

Independent loading and unloading test data can be provided in each of the three component directions. The components correspond to the response along the fill yarn, the response along the warp yarn, and the shear response.

Input File Usage: Use the following option to define the response along the fill yarn direction:

∗UNIAXIAL, COMPONENT=1

Use the following option to define the response along the warp yarn direction:

∗UNIAXIAL, COMPONENT=2

Use the following option to define the shear response:

∗UNIAXIAL, COMPONENT=SHEAR

Defining the deformation direction

The test data can be defined separately for tension and compression by specifying the deformation direction. If the deformation direction is defined (tension or compression), the tabular values defining tensile or compressive behavior should be specified with positive values of the stress and strain in the specified component and the loading data must start at the origin. If the behavior is not defined in a loading direction, the stress response will be zero in that direction (the fabric has no resistance in that direction).

If the deformation direction is not defined, the data apply to both tension and compression. However, the behavior is then considered to be nonlinear elastic and no unloading response can be specified. The test data will be considered to be symmetric about the origin if either tensile or compressive data are omitted.

Input File Usage: Use the following option to define tensile behavior:

∗LOADING DATA, DIRECTION=TENSION

Use the following option to define compressive behavior:

∗LOADING DATA, DIRECTION=COMPRESSION

Use the following option to define both tensile and compressive behavior in a single table:

∗LOADING DATA

Compressive behavior

In general, a fabric material does not have significant stiffness under compression. To prevent the collapse of wrinkled elements under compressive loading, the specified stress-strain curve should reinstate the compressive stiffness after a range of zero or very small resistance.

Defining loading/unloading component-wise response for a fabric material

To define the loading response, you specify the fabric stress as nonlinear functions of the fabric strain. This function can also depend on temperature and field variables. See "Input syntax rules," Section 1.2.1, for further information about defining data as functions of temperature and field variables.

The unloading response can be defined in the following different ways:

- You can specify several unloading curves that express the fabric stress as nonlinear functions of the fabric strain; Abaqus interpolates these curves to create an unloading curve that passes through the point of unloading in an analysis.
- You can specify an energy dissipation factor (and a permanent deformation factor for models with permanent deformation), from which Abaqus calculates a quadratic unloading function.
- You can specify an energy dissipation factor (and a permanent deformation factor for models with permanent deformation), from which Abaqus calculates an exponential unloading function.
- You can specify the fabric stress as a nonlinear function of the fabric strain, as well as a transition slope; the fabric unloads along the specified transition slope until it intersects the specified unloading function, at which point it unloads according to the function. (This unloading definition is referred to as combined unloading.)
- You can specify the fabric stress as a nonlinear function of the fabric strain; Abaqus shifts the specified unloading function along the strain axis so that it passes through the point of unloading in an analysis.

The behavior type that is specified for the fabric dictates the type of unloading you can define, as summarized in Table 23.4.1–1. The different behavior types, as well as the associated loading and unloading curves, are discussed in more detail in the sections that follow.

Defining nonlinear elastic behavior

The elastic behavior can be nonlinear and, optionally, rate dependent. When the loading response is rate dependent, a separate unloading curve must also be specified. However, the unloading response need not be rate dependent.

Defining rate-independent elasticity

When the loading response is rate independent, the unloading response is also rate independent and occurs along the same user-specified loading curve as illustrated in Figure 23.4.1–2. An unloading curve does not need to be specified.

Input File Usage: *LOADING DATA, TYPE=ELASTIC

Table 23.4.1–1 Available unloading definitions for the fabric behavior types.

Material behavior type	Unloading definition				
	Interpolated	Quadratic	Exponential	Combined	Shifted
Nonlinear elastic (rate-dependent only)	√				
Damaged elastic	√	√	√	√	
Permanent deformation	√	√	√		√

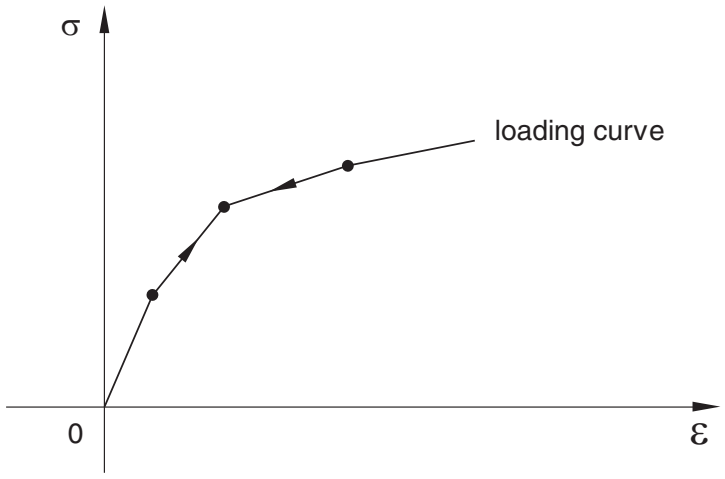


Figure 23.4.1–2 Nonlinear elastic rate-independent loading.

Defining rate-dependent elasticity

When the elastic response is rate dependent, both the loading and the unloading curves need to be specified. If the unloading data are not specified, the unloading occurs along the loading curve specified with the smallest rate of deformation.

Unphysical jumps in the stress due to sudden changes in the rate of deformation are prevented using a technique based on viscoplastic regularization. This technique also helps model relaxation effects in a very simplistic manner, with the relaxation time given as $\tau = \mu_0 + \mu_1|\lambda - 1|^{\alpha}$, where μ_0, μ_1, and α are material parameters and λ is the stretch. μ_0 is a linear viscosity parameter that controls the relaxation time when $\lambda \approx 1$. Small values of this parameter should be used; a suggested value is 0.0001s. μ_1 is a nonlinear viscosity parameter that controls the relaxation time at higher values of λ. The smaller

this value, the shorter the relaxation time. The suggested value for this parameter is 0.005s. α controls the sensitivity of the relaxation speed to the fabric strain component. Figure 23.4.1–3 illustrates the loading/unloading behavior as the component is loaded at a rate $\dot{\varepsilon}_2^l$ and then unloaded at a rate $\dot{\varepsilon}_2^u$.

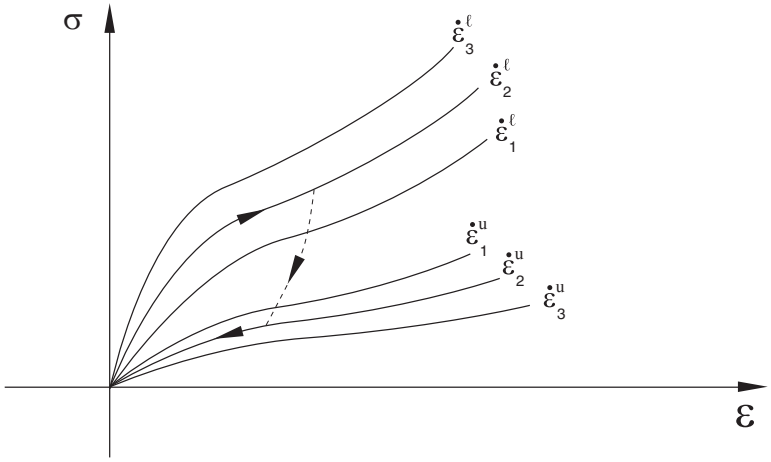


Figure 23.4.1–3 Rate-dependent loading/unloading.

The unloading path is determined by interpolating the specified unloading curves. The unloading need not be rate dependent, even though the loading response is rate dependent. When the unloading is rate dependent, the unloading path at any given component strain and strain rate is determined by interpolating the specified unloading curves.

Input File Usage: Use the following options when the unloading is also rate dependent:

∗LOADING DATA, TYPE=ELASTIC, RATE DEPENDENT, DIRECTION
∗UNLOADING DATA, DEFINITION=INTERPOLATED CURVE, RATE DEPENDENT

Use the following options when the unloading is rate independent:

∗LOADING DATA, TYPE=ELASTIC, RATE DEPENDENT, DIRECTION
∗UNLOADING DATA, DEFINITION=INTERPOLATED CURVE

Defining models with damage

The damage models dissipate energy upon unloading, and there is no permanent deformation upon complete unloading. You can specify the onset of damage by defining the strain above which the material response in unloading does not retrace the loading curve.

The unloading behavior controls the amount of energy dissipated by damage mechanisms and can be specified in one of the following ways:

- an analytical unloading curve (exponential/quadratic);
- an unloading curve interpolated from multiple user-specified unloading curves; or
- unloading along a transition unloading curve (constant slope specified by user) to the user-specified unloading curve (combined unloading).

Input File Usage: Use the following options to define damage with quadratic unloading behavior:

∗LOADING DATA, TYPE=DAMAGE, DIRECTION
∗UNLOADING DATA, DEFINITION=QUADRATIC

Use the following options to define damage with exponential unloading behavior:

∗LOADING DATA, TYPE=DAMAGE, DIRECTION
∗UNLOADING DATA, DEFINITION=EXPONENTIAL

Use the following options to define damage with an interpolated unloading curve:

∗LOADING DATA, TYPE=DAMAGE, DIRECTION
∗UNLOADING DATA, DEFINITION=INTERPOLATED CURVE

Use the following options to specify damage with combined unloading behavior:

∗LOADING DATA, TYPE=DAMAGE, DIRECTION
∗UNLOADING DATA, DEFINITION=COMBINED

Defining onset of damage

You can specify the onset of damage by defining the strain above which the material response in unloading does not retrace the loading curve.

Input File Usage: ∗LOADING DATA, TYPE=DAMAGE, DAMAGE ONSET=*value*

Specifying exponential/quadratic unloading

The damage model in Figure 23.4.1–4 is based on an analytical unloading curve that is derived from an energy dissipation factor, H (fraction of energy that is dissipated at any strain level). As the fabric component is loaded, the stress follows the path given by the loading curve. If the fabric component is unloaded (for example, at point ***B***), the stress follows the unloading curve BCO. Reloading after unloading follows the unloading curve OCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve. The arrows shown in Figure 23.4.1–4 illustrate the loading/unloading paths of this model.

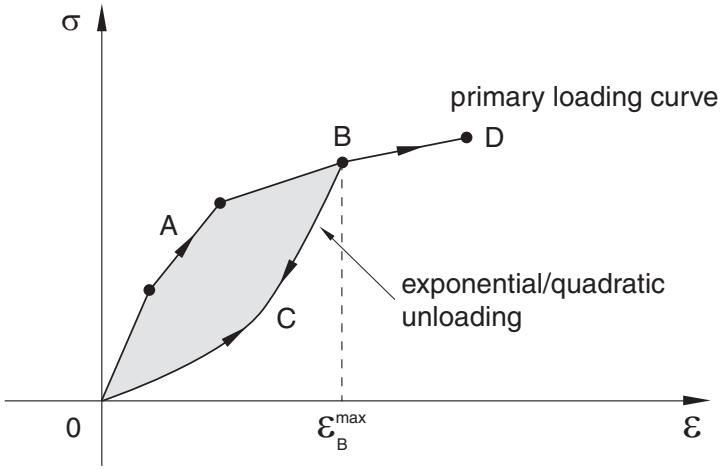


Figure 23.4.1–4 Exponential/quadratic unloading.

The unloading response follows the loading curve when the calculated unloading curve lies above the loading curve to prevent energy generation and follows a zero stress response when the unloading curve yields a negative response. In such cases the dissipated energy will be less than the value specified by the energy dissipation factor.

Specifying interpolated curve unloading

The damage model in Figure 23.4.1–5 illustrates an interpolated unloading response based on multiple unloading curves that intersect the primary loading curve at increasing values of stress/strains. You can specify as many unloading curves as are necessary to define the unloading response. Each unloading curve always starts at point *O*, the point of zero stress and zero strains, since the damage models do not allow any permanent deformation. The unloading curves are stored in normalized form so that they intersect the loading curve at a unit stress for a unit strain, and the interpolation occurs between these normalized curves. If unloading occurs from a maximum strain for which an unloading curve is not specified, the unloading is interpolated from neighboring unloading curves. As the fabric component is loaded, the stress follows the path given by the loading curve. If the fabric is unloaded (for example, at point *B*), the stress follows the unloading curve BCO. Reloading after unloading follows the unloading path OCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Specifying combined unloading

As illustrated in Figure 23.4.1–6, you can specify an unloading curve OCE in addition to the loading curve $OABD$ as well as a constant transition slope that connects the loading curve to the unloading

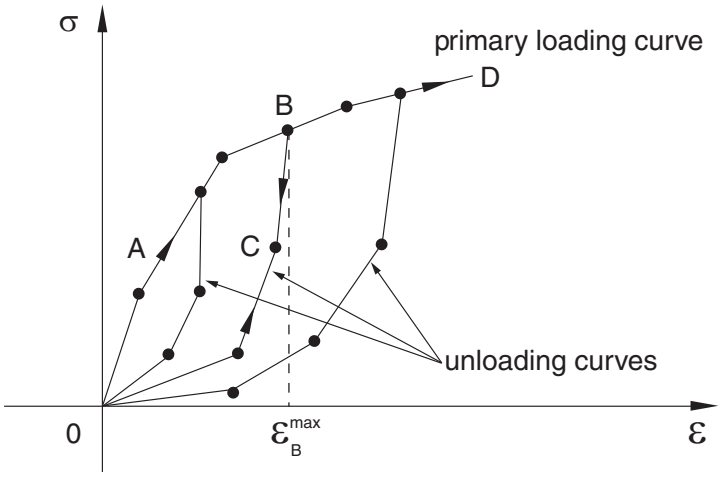


Figure 23.4.1–5 Interpolated curve unloading.

curve. As the fabric is loaded, the stress follows the path given by the loading curve. If the fabric is unloaded (for example at point ***B***) the stress follows the unloading curve BCO. The path BC is defined by the constant transition slope, and CO lies on the specified unloading curve. Reloading after unloading follows the unloading path OCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve.

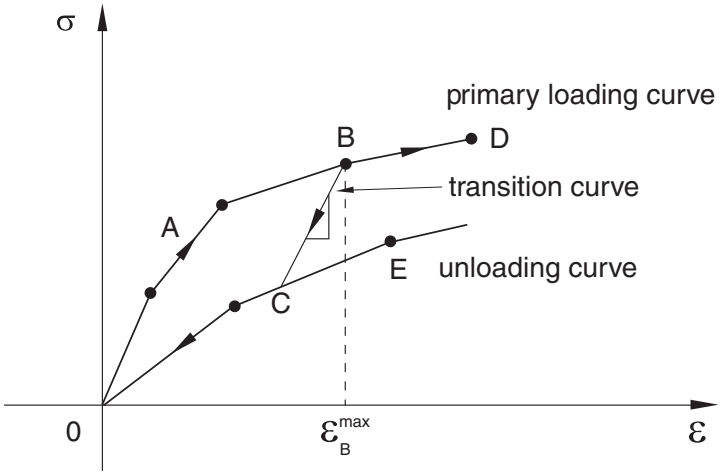


Figure 23.4.1–6 Combined unloading.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Defining models with permanent deformation

These models dissipate energy upon unloading and exhibit permanent deformation upon complete unloading. You can specify the onset of permanent deformation by defining the strain below which unloading occurs along the loading curve.

The unloading behavior controls the amount of energy dissipated as well as the amount of permanent deformation. The unloading behavior can be specified in one of the following ways:

- an analytical unloading curve (exponential/quadratic);
- an unloading curve interpolated from multiple user-specified unloading curves; or
- an unloading curve obtained by shifting the user-specified unloading curve to the point of unloading.

Input File Usage: Use the following options to define permanent deformation with quadratic unloading behavior:

```
*LOADING DATA, TYPE=PERMANENT DEFORMATION,
DIRECTION
*UNLOADING DATA, DEFINITION=QUADRATIC
```

Use the following options to define permanent damage with exponential unloading behavior:

```
*LOADING DATA, TYPE=PERMANENT DEFORMATION,
DIRECTION
*UNLOADING DATA, DEFINITION=EXPONENTIAL
```

Use the following options to define permanent damage with an interpolated unloading curve:

```
*LOADING DATA, TYPE=PERMANENT DEFORMATION,
DIRECTION
*UNLOADING DATA, DEFINITION=INTERPOLATED CURVE
```

Use the following options to specify permanent damage with a shifted unloading curve:

```
*LOADING DATA, TYPE=PERMANENT DEFORMATION,
DIRECTION
*UNLOADING DATA, DEFINITION=SHIFTED CURVE
```

Defining the onset of permanent deformation

By default, the onset of yield will be obtained as soon as the slope of the loading curve decreases by 10% from the maximum slope recorded up to that point while traversing along the loading curve. To override the default method of determining the onset of yield, you can specify either a value for the decrease in slope of the loading curve other than the default value of 10% (slope drop = 0.1) or by defining the strain below which unloading occurs along the loading curve. If a slope drop is specified, the onset of yield will be obtained as soon as the slope of the loading curve decreases by the specified factor from the maximum slope recorded up to that point.

Input File Usage: Use the following options to specify the onset of yield by defining the strain below which unloading occurs along the loading curve:

∗LOADING DATA, TYPE=PERMANENT DEFORMATION,
YIELD ONSET=*value*

Use the following options to specify the onset of yield by defining a slope drop for the loading curve:

∗LOADING DATA, TYPE=PERMANENT DEFORMATION,
SLOPE DROP=*value*

Specifying exponential/quadratic unloading

The model in Figure 23.4.1–7 illustrates an analytical unloading curve that is derived from an energy dissipation factor, H (fraction of energy that is dissipated at any strain level), and a permanent deformation factor, D_p. As the fabric component is loaded, the fabric stress follows the path given by the loading curve. If the component is unloaded (for example, at point ***B***), the stress follows the unloading curve BCD. The point ***D*** corresponds to the permanent deformation, $D_p\varepsilon_B^{max}$. Reloading after unloading follows the unloading curve DCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve. The arrows shown in Figure 23.4.1–7 illustrate the loading/unloading paths of this model.

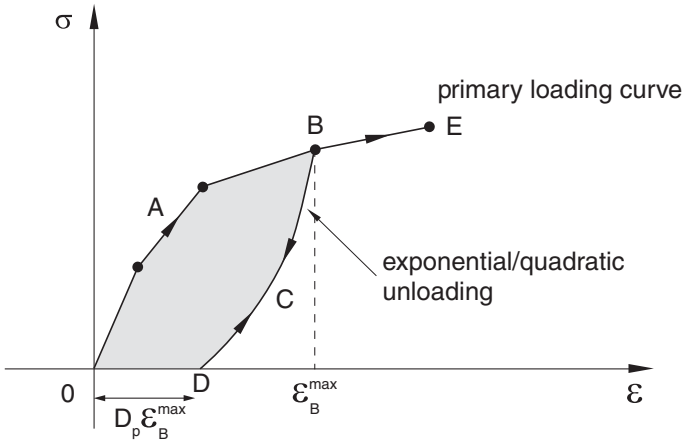


Figure 23.4.1–7 Exponential/quadratic unloading.

The unloading response follows the loading curve when the calculated unloading curve lies above the loading curve to prevent energy generation and follows a zero stress response when the unloading curve yields a negative response. In such cases the dissipated energy will be less than the value specified by the energy dissipation factor.

Specifying interpolated curve unloading

The model in Figure 23.4.1–8 illustrates an interpolated unloading response based on multiple unloading curves that intersect the primary loading curve at increasing values of stresses/strains.

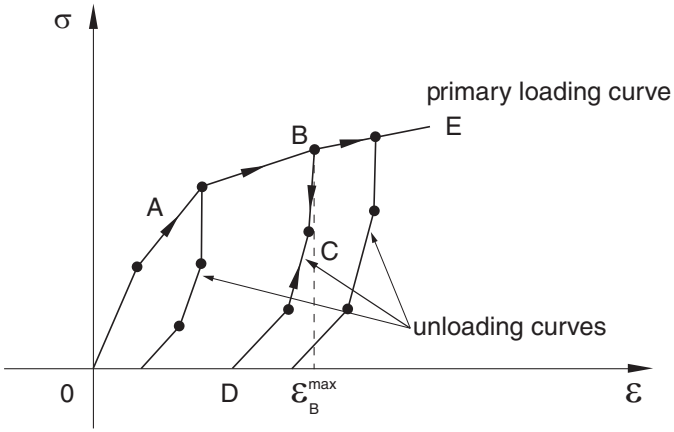


Figure 23.4.1–8 Interpolated curve unloading.

You can specify as many unloading curves as are necessary to define the unloading response. The first point of each unloading curve defines the permanent deformation if the fabric component is completely unloaded. The unloading curves are stored in normalized form so that they intersect the loading curve at a unit stress for a unit strain, and the interpolation occurs between these normalized curves. If unloading occurs from a maximum strain for which an unloading curve is not specified, the unloading is interpolated from neighboring unloading curves. As the fabric is loaded, the stress follows the path given by the loading curve. If the fabric is unloaded (for example, at point ***B***), the stress follows the unloading curve BCD. Reloading after unloading follows the unloading path DCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Specifying shifted curve unloading

You can specify an unloading curve passing through the origin in addition to the loading curve. The actual unloading curve is obtained by horizontally shifting the user-specified unloading curve to pass through the point of unloading as shown in Figure 23.4.1–9. The permanent deformation upon complete unloading is the horizontal shift applied to the unloading curve.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

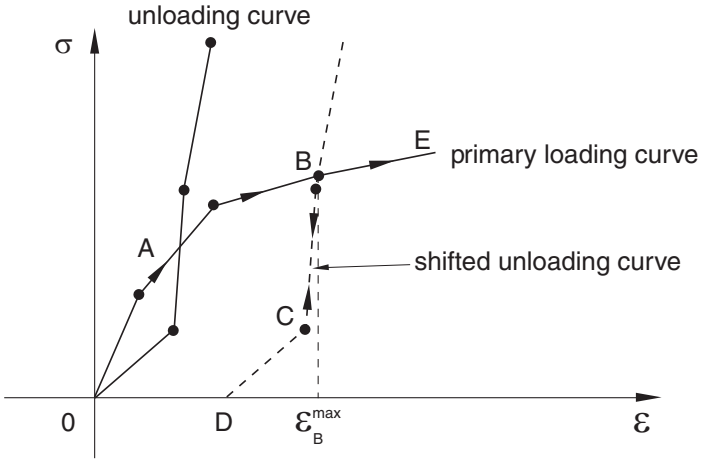


Figure 23.4.1–9 Shifted curve unloading.

Using different uniaxial models in tension and compression

When appropriate, different uniaxial behavior models can be used in tension and compression. For example, response under tension can be plastic with exponential unloading, while the response in compression can be nonlinear elastic (see Figure 23.4.1–10).

User-defined fabric materials

The mechanical response of a fabric material depends on a number of micro and meso-scale parameters covering the fabric construction and that of the individual yarns as a bundle of fibers. Often a multi-scale model becomes necessary to track the state of the fabric and its response to loading. Abaqus provides a specialized user subroutine, **VFABRIC**, to capture the complex fabric response given the deformed yarn directions and the strains along these directions.

The density ("Density," Section 21.2.1) is required when using a fabric material.

Input File Usage: Use the following options to define a fabric material through user subroutine **VFABRIC**:

∗MATERIAL, NAME=*name*
∗FABRIC, USER
∗DENSITY

Properties for a user-defined fabric material

Any material constants that are needed in user subroutine **VFABRIC** must be specified as part of a user-defined fabric material definition. Abaqus can be used to compute the isotropic thermal expansion response under thermal loading, even as the remaining mechanical response is defined by the user

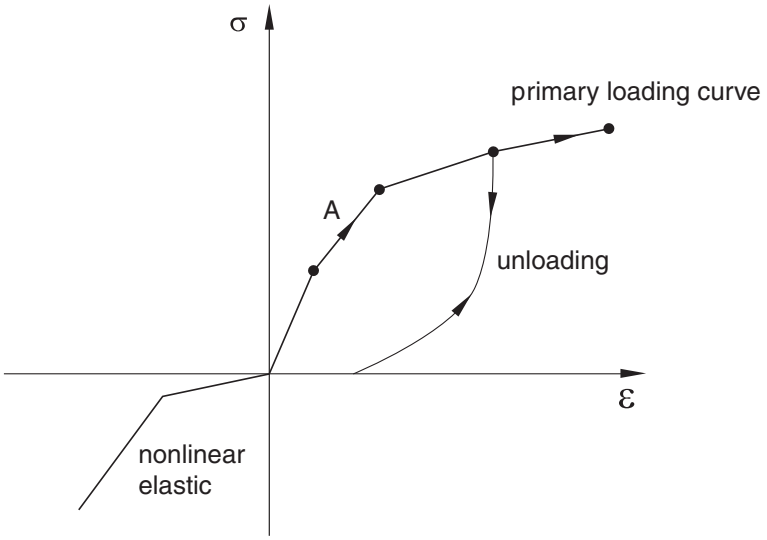


Figure 23.4.1–10 Different uniaxial models in tension and compression.

subroutine. Alternatively, you can include the thermal expansion within the definition of the mechanical response in user subroutine **VFABRIC**.

Input File Usage: Use the following option to define properties for a user-defined fabric material behavior:

∗FABRIC, USER, PROPERTIES=*number_of_constants*

Material state

Many mechanical constitutive models require the storage of solution-dependent state variables (plastic strains, "back stress," saturation values, etc. in rate constitutive forms or historical data for theories written in integral form). You should allocate storage for these variables in the associated material definition (see "Allocating space" in "User subroutines: overview," Section 18.1.1). There is no restriction on the number of state variables associated with a user-defined fabric material.

State variables associated with **VFABRIC** can be output to the output database (**.odb**) file and results (**.fil**) file using output identifiers SDV and SDV*n* (see "Abaqus/Explicit output variable identifiers," Section 4.2.2).

User subroutine **VFABRIC** is called for blocks of material points at each increment. When the subroutine is called, it is provided with the state at the start of the increment (fabric stress in the local system, solution-dependent state variables). It is also provided with the nominal fabric strains at the end of the increment and the incremental nominal fabric strains over the increment, both in the local system. The **VFABRIC** user material interface passes a block of material points to the subroutine on each call, which allows vectorization of the material subroutine.

The temperature is provided to user subroutine **VFABRIC** at the start and the end of the increment. The temperature is passed in as information only and cannot be modified, even in a fully coupled thermal-stress analysis. However, if the inelastic heat fraction is defined in conjunction with the specific heat and conductivity in a fully coupled thermal-stress analysis, the heat flux due to inelastic energy dissipation is calculated automatically. If user subroutine **VFABRIC** is used to define an adiabatic material behavior (conversion of plastic work to heat) in an explicit dynamic procedure, the temperatures must be stored and integrated as user-defined state variables. Most often the temperatures are provided by specifying initial conditions ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1) and are constant throughout the analysis.

Deleting elements from an Abaqus/Explicit mesh using state variables

Element deletion in a mesh can be controlled during the course of an Abaqus/Explicit analysis through user subroutine **VFABRIC**. Deleted elements have no ability to carry stresses and, therefore, have no contribution to the stiffness of the model. You specify the state variable number controlling the element deletion flag. For example, specifying a state variable number of 4 indicates that the fourth state variable is the deletion flag in **VFABRIC**. The deletion state variable should be set to a value of one or zero in **VFABRIC**. A value of one indicates that the material point is active, while a value of zero indicates that Abaqus/Explicit should delete the material point from the model by setting the stresses to zero. The structure of the block of material points passed to user subroutine **VFABRIC** remains unchanged during the analysis; deleted material points are not removed from the block. Abaqus/Explicit will pass zero stresses and strain increments for all deleted material points. Once a material point has been flagged as deleted, it cannot be reactivated. An element will be deleted from the mesh only after all of the material points in the element are deleted. The status of an element can be determined by requesting output of the variable STATUS. This variable is equal to 1 if the element is active and equal to 0 if the element is deleted.

Input File Usage: ∗DEPVAR, DELETE=*variable number*

Thermal expansion

You can define isotropic thermal expansion to specify the same coefficient of thermal expansion for the membrane and thickness-direction behaviors.

The membrane thermal strains, ε^{th}, are obtained as explained in "Thermal expansion," Section 26.1.2.

The elastic stretch in a given direction, $\lambda^{e\ell}$, relates the total stretch, λ, and the thermal stretch, λ^{th}:

$$\lambda^{e\ell} = \frac{\lambda}{\lambda^{th}}.$$

λ^{th} is given by

$$\lambda^{th} = 1 + \varepsilon^{th},$$

where ε^{th} is the linear thermal expansion strain in that direction.

Fabric thickness

The thickness of a fabric is difficult to measure experimentally. Fortunately, an accurate value for thickness is not always required due to the fact that a nominal stress measure, defined as force per unit area in the reference configuration, is used to characterize the in-plane response. An initial thickness can be specified on the section definition. Accurate tracking of the thickness with deformation is necessary only if the material is used with shell elements and the bending response needs to be captured accurately. You can compute the thickness direction strain increment when the fabric is defined through user subroutine **VFABRIC**. For test data–based fabric materials the thickness is assumed to remain constant with deformation. For a test data–based fabric definition, you must use the thickness value specified on the section definition for converting the experimental load data (which are typically available as force applied per unit width of the fabric) to stress quantities.

Defining a reference mesh (initial metric)

Abaqus/Explicit allows the specification of a reference mesh (initial metric) for fabrics modeled with membrane elements. For example, this is useful in airbag simulations to model wrinkles and changes in yarn orientations that arise from the airbag folding process. A flat mesh may be suitable for the unstressed reference configuration, but the initial state may require a corresponding folded mesh defining the folded state. The angular orientation of the yarn in the reference configuration is updated to obtain the new orientation in the initial configuration.

Input File Usage: Use the following option to define the reference configuration giving the element number and its nodal coordinates in the reference configuration:

*INITIAL CONDITIONS, TYPE=REF COORDINATE

Use the following option to define the reference configuration giving the node number and its coordinates in the reference configuration:

*INITIAL CONDITIONS, TYPE=NODE REF COORDINATE

Yarn behavior under initial compressive strains

Defining a reference configuration that is different from the initial configuration generally results in nonzero stresses and strains in the initial configuration based on the material definition. By default, compressive initial strains in the yarn directions generate zero stresses. The stress remains zero as the strain is continuously recovered from the initial compressive values toward the strain-free state. Once the initial slack is recovered, any subsequent compressive/tensile strains generate stresses as per the material definition.

Input File Usage: Use the following option to specify that initial compressive strains are recovered stress free (default):

*FABRIC, STRESS FREE INITIAL SLACK=YES

Use the following option to specify that initial compressive strains generate nonzero initial stresses:

*FABRIC, STRESS FREE INITIAL SLACK=NO

Defining yarn directions in the reference configuration

In general, the yarn directions may not be orthogonal to each other in the reference configuration. You can specify these local directions with respect to the in-plane axes of an orthogonal orientation system at a material point. Both the local directions and the orthogonal system are defined together as a single orientation definition. See "Orientations," Section 2.2.5, for more information.

If the local directions are not specified, these directions are assumed to match the in-plane axes of the orthogonal system defined. The local direction may not remain orthogonal with deformation. Abaqus updates the local directions with deformation and computes the nominal strains along these directions and the angle between them (fabric shear strain). The constitutive behavior for the fabric defines the nominal stresses in the local system in terms of the fabric strain.

Local yarn directions can be output to the output database as described in "Output," below.

Picture-frame shear fabric test

The shear response of the fabric is typically studied using a picture-frame shear test. The reference and the deformed configuration for a picture-frame shear test under force F is illustrated in Figure 23.4.1–11, where L_0 is the size of the picture-frame, and ψ_{12}^0 is the initial angle between the yarn directions. The four sides of the specimen are constrained not to change in their length even as the frame elongates and the angle between the yarn directions ψ_{12} decreases with deformation.The relationship between the nominal shear stress, T_{12}, and the applied force, F, is

$$T_{12} = \left(\frac{FL_0}{v_0}\right) \sin\left(\frac{\psi_{12}}{2}\right),$$

where v_0 is the initial volume of the specimen. The fabric engineering shear strain, γ_{12}, is related to the change in the angle between the yarn directions as

$$\gamma_{12} = \psi_{12}^0 - \psi_{12}.$$

Use with other material models

The fabric material model can be used by itself, or it can be combined with isotropic thermal expansion to introduce thermal volume changes ("Thermal expansion," Section 26.1.2). See "Combining material behaviors," Section 21.1.3, for more details. Thermal expansion can alternatively be an integral part of the constitutive model implemented in **VFABRIC** for user-defined fabric materials.

For a test-data based fabric material, both the mass proportional and the stiffness proportional damping can be specified (see "Material damping," Section 26.1.1). If stiffness proportional damping is specified, Abaqus calculates the damping stress based on the current elastic stiffness of the material and the resulting damping stress is included in the reported stress output at the integration points.

For a fabric material defined by user subroutine **VFABRIC**, mass proportional damping can be specified, but stiffness proportional damping must be defined within the user subroutine.

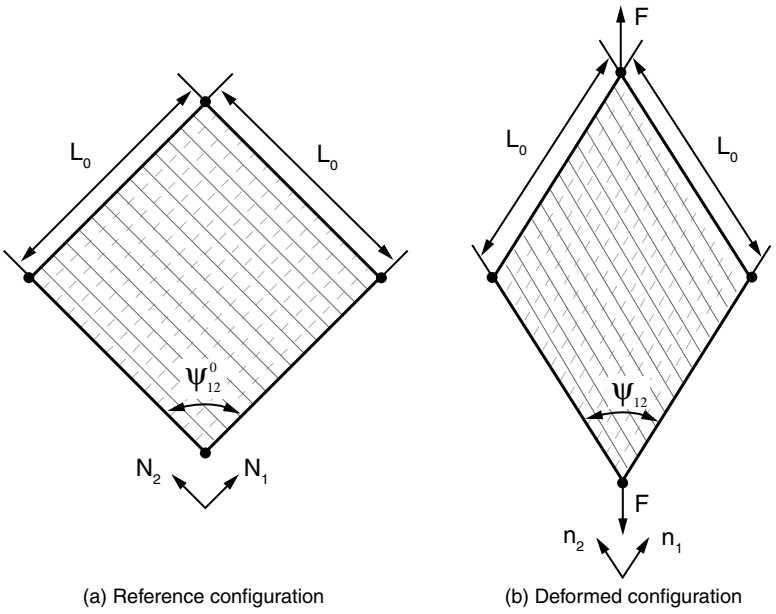


Figure 23.4.1–11 Picture-frame shear test on a fabric.

Elements

The fabric material model can be used with plane stress elements (plane stress solid elements, finite-strain shells, and membranes). It is recommended that the fabric material model be used with fully integrated or triangular membrane elements. When the fabric material model is used with shell elements, Abaqus does not compute a default transverse shear stiffness and you must specify it directly (see"Defining the transverse shear stiffness" in "Using a shell section integrated during the analysis to define the section behavior," Section 29.6.5).

Procedures

Fabric materials must always be used with geometrically nonlinear analyses ("General and linear perturbation procedures," Section 6.1.3).

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables have special meaning for the fabric material models:

EFABRIC Nominal fabric strain with components similar to that of LE, but with the direct components measuring the nominal strain along the yarn directions and the engineering shear component measuring the change in angle between the two yarn directions.

SFABRIC Nominal fabric stress with components similar to that of the regular Cauchy stress, S, but with the direct components measuring the nominal stress along the yarn directions and the shear component measuring response to the change in angle between the two yarn directions.

By default Abaqus outputs local material directions whenever element field output is requested to the output database for fabric models. The local directions are output as field variables (LOCALDIR1, LOCALDIR2, LOCALDIR3) representing the yarn direction cosines; these variables can be visualized as vector plots in the Visualization module of Abaqus/CAE (Abaqus/Viewer).

Output of local material directions is suppressed if no element field output is requested or if you specify not to have element material directions written to the output database (see “Specifying the directions for element output in Abaqus/Standard and Abaqus/Explicit” in “Output to the output database,” Section 4.1.3).

23.5 Jointed materials

- “Jointed material model,” Section 23.5.1

23.5.1 JOINTED MATERIAL MODEL

Product: Abaqus/Standard

References

- "Orientations," Section 2.2.5
- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *JOINTED MATERIAL

Overview

The jointed material model:

- is intended to provide a simple continuum model for a material containing a high density of parallel joint surfaces where each system of parallel joints is associated with a particular orientation, such as sedimentary rock;
- assumes that the spacing of the joints of a particular orientation is sufficiently close compared to characteristic dimensions in the domain of the model such that the joints can be smeared into a continuum of slip systems;
- provides for opening or frictional sliding of the joints in each of these systems (a "system" in this context is a joint orientation in a particular direction at a material calculation point); and
- assumes that the elastic behavior of the material is isotropic and linear when all joints at a point are closed (isotropic linear elastic behavior must be included in the material definition; see "Defining isotropic elasticity" in "Linear elastic behavior," Section 22.2.1).

Joint opening/closing

The jointed material model is intended primarily for applications where the stresses are mainly compressive. The model provides a joint opening capability when the stress normal to the joint tries to become tensile. In this case the stiffness of the material normal to the joint plane becomes zero instantaneously. Abaqus/Standard uses a stress-based joint opening criterion, whereas joint closing is monitored based on strain. Joint system a opens when the estimated pressure stress across the joint (normal to the joint surface) is no longer positive:

$$p_a \leq 0.$$

In this case the material is assumed to have no elastic stiffness with respect to direct strain across the joint system. Open joints thus create anisotropic elastic response at a point. The joint system remains open so long as

$$\varepsilon^{el}_{an(ps)} \leq \varepsilon^{el}_{an},$$

where ε^{el}_{an} is the component of direct elastic strain across the joint and $\varepsilon^{el}_{an(ps)}$ is the component of direct elastic strain across the joint calculated in plane stress as

$$\varepsilon^{el}_{an(ps)} = -\frac{\nu}{E}(\sigma_{a1} + \sigma_{a2}),$$

where E is the Young's modulus of the material, ν is the Poisson's ratio, and σ_{a1}, σ_{a2} are the direct stresses in the plane of the joint.

The shear response of open joints is governed by the shear retention parameter, f_{sr}, which represents the fraction of the elastic shear modulus retained when the joints are open (f_{sr}=0 means no shear stiffness associated with open joints, while f_{sr}=1 corresponds to elastic shear stiffness in open joints; any value between these two extremes can be used). When a joint opens, the shear behavior may be brittle, depending on the shear retention factor used for open joints. In addition, the stiffness of the material normal to the joint plane suddenly goes to zero. For these reasons, in situations where the confining stresses are low or significant regions experience tensile behavior, the joint systems may experience a sequence of alternate opening and closing states from iteration to iteration. Typically such behavior manifests itself as oscillating global residual forces. The convergence rate associated with such discontinuous behavior may be very slow and, thus, prohibit obtaining a solution. This type of failure is more probable in cases where more than one joint system is modeled.

Improving convergence when joints open and close repeatedly

When the repeated opening and closing of joints makes convergence difficult, you can improve convergence by preventing a joint from opening. In this case an elastic stiffness is always associated with the joint. It is most useful when the opening and closing of joints is limited to small regions of the model. You can prevent a joint from opening only when the joint direction is specified, as described below.

Input File Usage: *JOINTED MATERIAL, NO SEPARATION, JOINT DIRECTION

Specifying nonzero shear retention in open joints

You must specify nonzero shear retention in open joints directly. The parameter f_{sr} can be defined as a tabular function of temperature and predefined field variables.

Input File Usage: *JOINTED MATERIAL, SHEAR RETENTION

Compressive joint sliding

The failure surface for sliding on joint system a is defined by

$$f_a = \tau_a - p_a \tan\beta_a - d_a = 0,$$

where τ_a is the magnitude of the shear stress resolved onto the joint surface, p_a is the normal pressure stress acting across the joint, β_a is the friction angle for system a, and d_a is the cohesion for system a. So

long as $f_a < 0$, joint system a does not slip. When $f_a = 0$, joint system a slips. The inelastic ("plastic") strain on the system is given by

$$d\gamma_{a\alpha}^{pl} = d\bar{\varepsilon}_a^{pl} \frac{\tau_{a\alpha}}{\tau_a} \cos\psi_a$$
$$d\varepsilon_{an}^{pl} = d\bar{\varepsilon}_a^{pl} \sin\psi_a,$$

where

$d\gamma_{a\alpha}^{pl}$ is the rate of inelastic shear strain in direction α on the joint surface ($\alpha = 1, 2$ are orthogonal directions on the joint surface),

$d\bar{\varepsilon}_a^{pl}$ is the magnitude of the inelastic strain rate,

$\tau_{a\alpha}$ is a component of the shear stress on the joint surface,

ψ_a is the dilation angle for this joint system (choosing $\psi_a = 0$ provides pure shear flow on the joint, while $\psi_a > 0$ causes dilation of the joint as it slips), and

$d\varepsilon_{an}^{pl}$ is the inelastic strain normal to the joint surface.

The sliding of the different joint systems at a point is independent, in the sense that sliding on one system does not change the failure criterion or the dilation angle for any other joint system at the same point.

Up to three joint directions can be included in the material description. The orientations of the joint directions are given by referring to the names of user-defined local orientations ("Orientations," Section 2.2.5) that define the joint orientations in the original configuration. Output of stress and strain components is in the global directions unless a local orientation is also used in the material's section definition.

The parameters β_a, ψ_a, and d_a can be specified as tabular functions of temperature and/or predefined field variables for each joint direction.

Input File Usage: Use both of the following options:

∗ORIENTATION, NAME=*name*
∗JOINTED MATERIAL, JOINT DIRECTION=*name*

Repeat the ∗JOINTED MATERIAL option for each direction to be specified, up to three times.

Joint directions and finite rotations

In geometrically nonlinear analysis steps the joint directions always remain fixed in space.

Bulk failure

In addition to the joint systems, the jointed material model includes a bulk material failure mechanism, which is based on the Drucker-Prager failure criterion:

$$q - p\tan\beta_b - d_b = 0,$$

where

$q \stackrel{\text{def}}{=} \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$	is the Mises equivalent deviatoric stress,
$\mathbf{S} \stackrel{\text{def}}{=} \boldsymbol{\sigma} + p\mathbf{I}$	is the deviatoric stress,
$p \stackrel{\text{def}}{=} -\frac{1}{3}\mathbf{I} : \boldsymbol{\sigma}$	is the equivalent pressure stress,
β_b	is the friction angle for the bulk material, and
d_b	is the cohesion for the bulk material.

If this failure criterion is reached, the bulk inelastic flow is defined by

$$d\boldsymbol{\varepsilon}_b^{pl} = d\bar{\varepsilon}_b^{pl} \frac{1}{1 - \frac{1}{3}\tan\psi_b} \frac{\partial g_b}{\partial \boldsymbol{\sigma}},$$

where

$$g_b = q - p\tan\psi_b$$

is the flow potential. Here $d\bar{\varepsilon}_b^{pl}$ is the magnitude of the inelastic flow rate (chosen so that $d\bar{\varepsilon}_b^{pl} = |(d\varepsilon_b^{pl})_{11}|$ in uniaxial compression in the 1-direction), and ψ_b is the dilation angle for the bulk material. This bulk failure model is a simplified version of the extended Drucker-Prager model ("Extended Drucker-Prager models," Section 23.3.1). This bulk failure system is independent of the joint systems in that bulk inelastic flow does not change the behavior of any joint system.

If bulk material failure is to be modeled, a jointed material behavior must be specified to define the parameters associated with bulk material failure behavior. Thus, up to five jointed material behaviors can appear in the same material definition: three joint directions, shear retention in open joints, and bulk material failure.

The parameters β_b, ψ_b, and d_b can be specified as a tabular function of temperature and/or predefined field variables.

Input File Usage: *JOINTED MATERIAL (the JOINT DIRECTION parameter must be omitted)

Nonassociated flow

If $\psi \neq \beta$ in any joint system, whether it be associated with the joint surfaces or the bulk material, the flow in that system is "nonassociated." The implication is that the material stiffness matrix is not symmetric. Therefore, the unsymmetric matrix solution scheme should be used for the analysis step ("Defining an analysis," Section 6.1.2), especially when large regions of the model are expected to flow plastically and when the difference between ψ and β is large. If the difference between ψ and β is not large, a symmetric approximation to the matrix can provide an acceptable rate of convergence of the equilibrium equations and, hence, a lower overall solution cost. Therefore, the unsymmetric matrix solution scheme is not invoked automatically when jointed material behavior is defined.

Elements

The jointed material model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus/Standard. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

23.6 Concrete

- “Concrete smeared cracking,” Section 23.6.1
- “Cracking model for concrete,” Section 23.6.2
- “Concrete damaged plasticity,” Section 23.6.3

23.6.1 CONCRETE SMEARED CRACKING

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *CONCRETE
- *TENSION STIFFENING
- *SHEAR RETENTION
- *FAILURE RATIOS
- "Defining concrete smeared cracking" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The smeared crack concrete model in Abaqus/Standard:

- provides a general capability for modeling concrete in all types of structures, including beams, trusses, shells, and solids;
- can be used for plain concrete, even though it is intended primarily for the analysis of reinforced concrete structures;
- can be used with rebar to model concrete reinforcement;
- is designed for applications in which the concrete is subjected to essentially monotonic straining at low confining pressures;
- consists of an isotropically hardening yield surface that is active when the stress is dominantly compressive and an independent "crack detection surface" that determines if a point fails by cracking;
- uses oriented damaged elasticity concepts (smeared cracking) to describe the reversible part of the material's response after cracking failure;
- requires that the linear elastic material model (see "Linear elastic behavior," Section 22.2.1) be used to define elastic properties; and
- cannot be used with local orientations (see "Orientations," Section 2.2.5).

See "Inelastic behavior," Section 23.1.1, for a discussion of the concrete models available in Abaqus.

Reinforcement

Reinforcement in concrete structures is typically provided by means of rebars, which are one-dimensional strain theory elements (rods) that can be defined singly or embedded in oriented surfaces. Rebars

are typically used with metal plasticity models to describe the behavior of the rebar material and are superposed on a mesh of standard element types used to model the concrete.

With this modeling approach, the concrete behavior is considered independently of the rebar. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, are modeled approximately by introducing some "tension stiffening" into the concrete modeling to simulate load transfer across cracks through the rebar. Details regarding tension stiffening are provided below.

Defining the rebar can be tedious in complex problems, but it is important that this be done accurately since it may cause an analysis to fail due to lack of reinforcement in key regions of a model. See "Defining reinforcement," Section 2.2.3, for more information regarding rebars.

Cracking

The model is intended as a model of concrete behavior for relatively monotonic loadings under fairly low confining pressures (less than four to five times the magnitude of the largest stress that can be carried by the concrete in uniaxial compression).

Crack detection

Cracking is assumed to be the most important aspect of the behavior, and representation of cracking and of postcracking behavior dominates the modeling. Cracking is assumed to occur when the stress reaches a failure surface that is called the "crack detection surface." This failure surface is a linear relationship between the equivalent pressure stress, p, and the Mises equivalent deviatoric stress, q, and is illustrated in Figure 23.6.1–5. When a crack has been detected, its orientation is stored for subsequent calculations. Subsequent cracking at the same point is restricted to being orthogonal to this direction since stress components associated with an open crack are not included in the definition of the failure surface used for detecting the additional cracks.

Cracks are irrecoverable: they remain for the rest of the calculation (but may open and close). No more than three cracks can occur at any point (two in a plane stress case, one in a uniaxial stress case). Following crack detection, the crack affects the calculations because a damaged elasticity model is used. Oriented, damaged elasticity is discussed in more detail in "An inelastic constitutive model for concrete," Section 4.5.1 of the Abaqus Theory Manual.

Smeared cracking

The concrete model is a smeared crack model in the sense that it does not track individual "macro" cracks. Constitutive calculations are performed independently at each integration point of the finite element model. The presence of cracks enters into these calculations by the way in which the cracks affect the stress and material stiffness associated with the integration point.

Tension stiffening

The postfailure behavior for direct straining across cracks is modeled with tension stiffening, which allows you to define the strain-softening behavior for cracked concrete. This behavior also allows for the effects of the reinforcement interaction with concrete to be simulated in a simple manner. Tension

stiffening is required in the concrete smeared cracking model. You can specify tension stiffening by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion.

Postfailure stress-strain relation

Specification of strain softening behavior in reinforced concrete generally means specifying the postfailure stress as a function of strain across the crack. In cases with little or no reinforcement this specification often introduces mesh sensitivity in the analysis results in the sense that the finite element predictions do not converge to a unique solution as the mesh is refined because mesh refinement leads to narrower crack bands. This problem typically occurs if only a few discrete cracks form in the structure, and mesh refinement does not result in formation of additional cracks. If cracks are evenly distributed (either due to the effect of rebar or due to the presence of stabilizing elastic material, as in the case of plate bending), mesh sensitivity is less of a concern.

In practical calculations for reinforced concrete, the mesh is usually such that each element contains rebars. The interaction between the rebars and the concrete tends to reduce the mesh sensitivity, provided that a reasonable amount of tension stiffening is introduced in the concrete model to simulate this interaction (Figure 23.6.1–1).

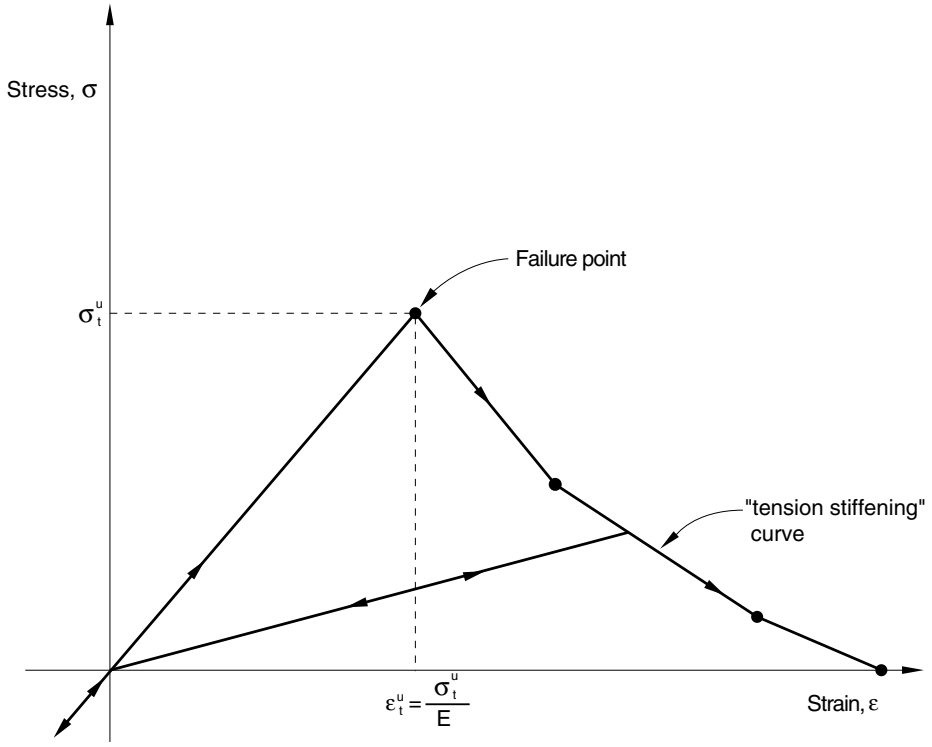


Figure 23.6.1–1 "Tension stiffening" model.

The tension stiffening effect must be estimated; it depends on such factors as the density of reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the mesh. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain of about 10 times the strain at failure. The strain at failure in standard concretes is typically 10^{-4}, which suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to a particular case.

The choice of tension stiffening parameters is important in Abaqus/Standard since, generally, more tension stiffening makes it easier to obtain numerical solutions. Too little tension stiffening will cause the local cracking failure in the concrete to introduce temporarily unstable behavior in the overall response of the model. Few practical designs exhibit such behavior, so that the presence of this type of response in the analysis model usually indicates that the tension stiffening is unreasonably low.

Input File Usage: Use both of the following options:

```
*CONCRETE
*TENSION STIFFENING, TYPE=STRAIN (default)
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Smeared Cracking**: **Suboptions→Tension Stiffening**: **Type: Strain**

Fracture energy cracking criterion

As discussed earlier, when there is no reinforcement in significant regions of a concrete model, the strain softening approach for defining tension stiffening may introduce unreasonable mesh sensitivity into the results. Crisfield (1986) discusses this issue and concludes that Hillerborg's (1976) proposal is adequate to allay the concern for many practical purposes. Hillerborg defines the energy required to open a unit area of crack as a material parameter, using brittle fracture concepts. With this approach the concrete's brittle behavior is characterized by a stress-*displacement* response rather than a stress-*strain* response. Under tension a concrete specimen will crack across some section. After it has been pulled apart sufficiently for most of the stress to be removed (so that the elastic strain is small), its length will be determined primarily by the opening at the crack. The opening does not depend on the specimen's length (Figure 23.6.1–2).

Implementation

The implementation of this stress-displacement concept in a finite element model requires the definition of a characteristic length associated with an integration point. The characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic crack length is used because the direction in which cracks will occur is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which

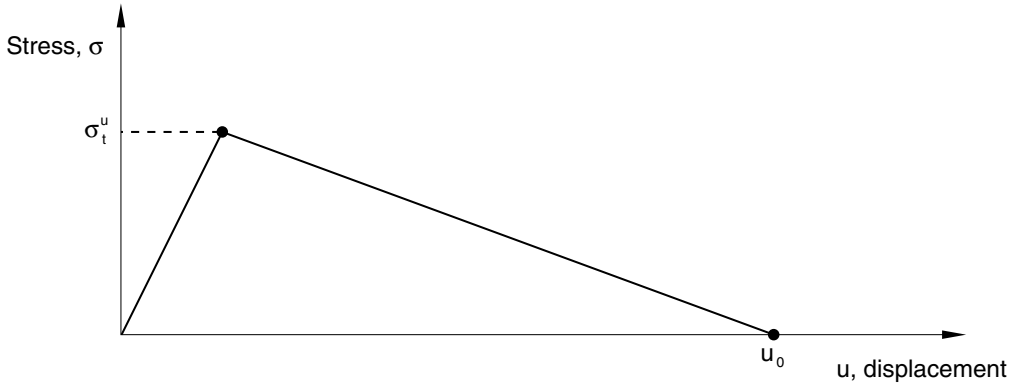


Figure 23.6.1–2 Fracture energy cracking model.

they crack: some mesh sensitivity remains because of this effect, and elements that are as close to square as possible are recommended.

This approach to modeling the concrete's brittle response requires the specification of the displacement u_0 at which a linear approximation to the postfailure strain softening gives zero stress (see Figure 23.6.1–2).

The failure stress, σ_t^u, occurs at a failure *strain* (defined by the failure stress divided by the Young's modulus); however, the stress goes to zero at an ultimate *displacement*, u_0, that is independent of the specimen length. The implication is that a displacement-loaded specimen can remain in static equilibrium after failure only if the specimen is short enough so that the strain at failure, ε_t^u, is less than the strain at this value of the displacement:

$$\varepsilon_t^u < u_0/L,$$

where L is the length of the specimen.

Input File Usage: Use both of the following options:

```
*CONCRETE
*TENSION STIFFENING, TYPE=DISPLACEMENT
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Smeared Cracking**: **Suboptions→Tension Stiffening**: **Type: Displacement**

Obtaining the ultimate displacement

The ultimate displacement, u_0, can be estimated from the fracture energy per unit area, G_f, as $u_0 = 2G_f/\sigma_t^u$, where σ_t^u is the maximum tensile stress that the concrete can carry. Typical values for u_0 are 0.05 mm (2×10^{-3} in) for a normal concrete to 0.08 mm (3×10^{-3} in) for a high strength concrete. A typical value for ε_t^u is about 10^{-4}, so that the requirement is that $L < 500$ mm (20 in).

Critical length

If the specimen is longer than the critical length, L, more strain energy is stored in the specimen than can be dissipated by the cracking process when it cracks under fixed displacement. Some of the strain energy must, therefore, be converted into kinetic energy, and the failure event must be dynamic even under prescribed displacement loading. This implies that, when this approach is used in finite elements, characteristic element dimensions must be less than this critical length, or additional (dynamic) considerations must be included. The analysis input file processor checks the characteristic length of each element using this concrete model and will not allow any element to have a characteristic length that exceeds u_0/ε_t^u. You must remesh with smaller elements where necessary or use the stress-strain definition of tension stiffening. Since the fracture energy approach is generally used only for plain concrete, this rarely places any limit on the meshing.

Cracked shear retention

As the concrete cracks, its shear stiffness is diminished. This effect is defined by specifying the reduction in the shear modulus as a function of the opening strain across the crack. You can also specify a reduced shear modulus for closed cracks. This reduced shear modulus will also have an effect when the normal stress across a crack becomes compressive. The new shear stiffness will have been degraded by the presence of the crack.

The modulus for shearing of cracks is defined as ϱG, where G is the elastic shear modulus of the uncracked concrete and ϱ is a multiplying factor. The shear retention model assumes that the shear stiffness of open cracks reduces linearly to zero as the crack opening increases:

$$\varrho = (1 - \varepsilon/\varepsilon^{\max}) \quad \text{for} \quad \varepsilon < \varepsilon^{\max}, \qquad \varrho = 0 \quad \text{for} \quad \varepsilon \geq \varepsilon^{\max},$$

where ε is the direct strain across the crack and $\varepsilon^{\max}$ is a user-specified value. The model also assumes that cracks that subsequently close have a reduced shear modulus:

$$\varrho = \varrho^{\text{close}} \quad \text{for} \quad \varepsilon < 0,$$

where you specify ϱ^{close}.

ϱ^{close} and $\varepsilon^{\max}$ can be defined with an optional dependency on temperature and/or predefined field variables. If shear retention is not included in the material definition for the concrete smeared cracking model, Abaqus/Standard will automatically invoke the default behavior for shear retention such that the shear response is unaffected by cracking (full shear retention). This assumption is often reasonable: in many cases, the overall response is not strongly dependent on the amount of shear retention.

Input File Usage: Use both of the following options:

```
*CONCRETE
*SHEAR RETENTION
```

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Smeared Cracking**: **Suboptions→Shear Retention**

Compressive behavior

When the principal stress components are dominantly compressive, the response of the concrete is modeled by an elastic-plastic theory using a simple form of yield surface written in terms of the equivalent pressure stress, p, and the Mises equivalent deviatoric stress, q; this surface is illustrated in Figure 23.6.1–5. Associated flow and isotropic hardening are used. This model significantly simplifies the actual behavior. The associated flow assumption generally over-predicts the inelastic volume strain. The yield surface cannot be matched accurately to data in triaxial tension and triaxial compression tests because of the omission of third stress invariant dependence. When the concrete is strained beyond the ultimate stress point, the assumption that the elastic response is not affected by the inelastic deformation is not realistic. In addition, when concrete is subjected to very high pressure stress, it exhibits inelastic response: no attempt has been made to build this behavior into the model.

The simplifications associated with compressive behavior are introduced for the sake of computational efficiency. In particular, while the assumption of associated flow is not justified by experimental data, it can provide results that are acceptably close to measurements, provided that the range of pressure stress in the problem is not large. From a computational viewpoint, the associated flow assumption leads to enough symmetry in the Jacobian matrix of the integrated constitutive model (the "material stiffness matrix") such that the overall equilibrium equation solution usually does not require unsymmetric equation solution. All of these limitations could be removed at some sacrifice in computational cost.

You can define the stress-strain behavior of plain concrete in uniaxial compression outside the elastic range. Compressive stress data are provided as a tabular function of plastic strain and, if desired, temperature and field variables. Positive (absolute) values should be given for the compressive stress and strain. The stress-strain curve can be defined beyond the ultimate stress, into the strain-softening regime.

Input File Usage: *CONCRETE

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Smeared Cracking**

Uniaxial and multiaxial behavior

The cracking and compressive responses of concrete that are incorporated in the concrete model are illustrated by the uniaxial response of a specimen shown in Figure 23.6.1–3.

When concrete is loaded in compression, it initially exhibits elastic response. As the stress is increased, some nonrecoverable (inelastic) straining occurs and the response of the material softens. An ultimate stress is reached, after which the material loses strength until it can no longer carry any stress. If the load is removed at some point after inelastic straining has occurred, the unloading response is softer than the initial elastic response: the elasticity has been damaged. This effect is ignored in the model, since we assume that the applications involve primarily monotonic straining, with only occasional, minor unloadings. When a uniaxial concrete specimen is loaded in tension, it responds elastically until, at a stress that is typically 7%–10% of the ultimate compressive stress, cracks form. Cracks form so quickly that, even in the stiffest testing machines available, it is very difficult to observe the actual behavior. The

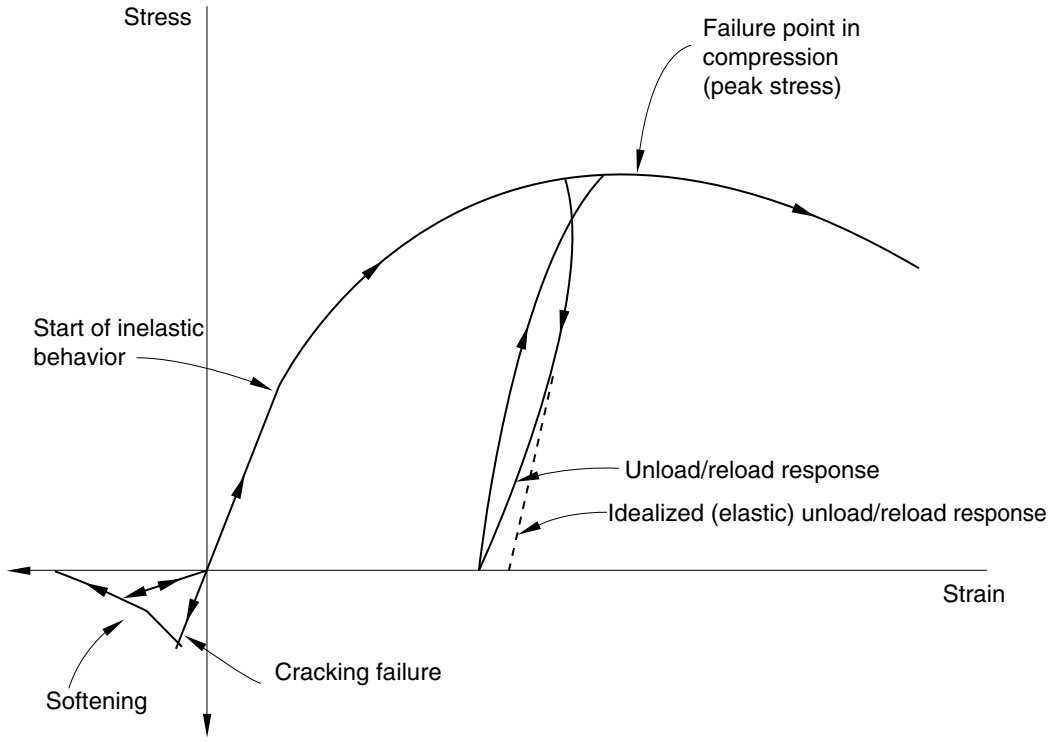


Figure 23.6.1–3 Uniaxial behavior of plain concrete.

model assumes that cracking causes damage, in the sense that open cracks can be represented by a loss of elastic stiffness. It is also assumed that there is no permanent strain associated with cracking. This will allow cracks to close completely if the stress across them becomes compressive.

In multiaxial stress states these observations are generalized through the concept of surfaces of failure and flow in stress space. These surfaces are fitted to experimental data. The surfaces used are shown in Figure 23.6.1–4 and Figure 23.6.1–5.

Failure surface

You can specify failure ratios to define the shape of the failure surface (possibly as a function of temperature and predefined field variables). Four failure ratios can be specified:

- The ratio of the ultimate biaxial compressive stress to the ultimate uniaxial compressive stress.
- The absolute value of the ratio of the uniaxial tensile stress at failure to the ultimate uniaxial compressive stress.
- The ratio of the magnitude of a principal component of plastic strain at ultimate stress in biaxial compression to the plastic strain at ultimate stress in uniaxial compression.

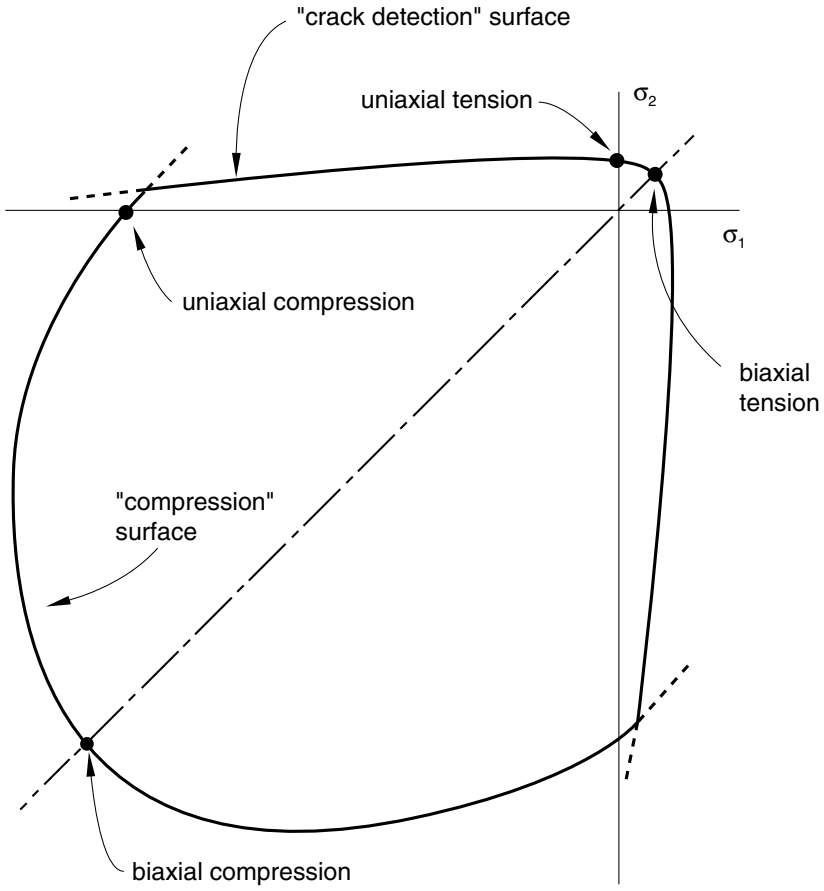


Figure 23.6.1–4 Yield and failure surfaces in plane stress.

- The ratio of the tensile principal stress at cracking, in plane stress, when the other principal stress is at the ultimate compressive value, to the tensile cracking stress under uniaxial tension.

Default values of the above ratios are used if you do not specify them.

Input File Usage: *FAILURE RATIOS

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Smeared Cracking**: **Suboptions→Failure Ratios**

Response to strain reversals

Because the model is intended for application to problems involving relatively monotonic straining, no attempt is made to include prediction of cyclic response or of the reduction in the elastic stiffness caused

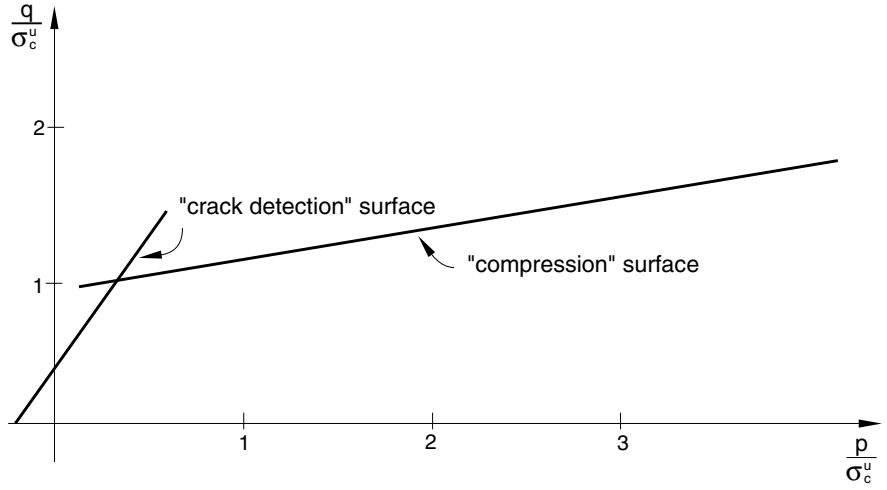


Figure 23.6.1–5 Yield and failure surfaces in the (p–q) plane.

by inelastic straining under predominantly compressive stress. Nevertheless, it is likely that, even in those applications for which the model is designed, the strain trajectories will not be entirely radial, so that the model should predict the response to occasional strain reversals and strain trajectory direction changes in a reasonable way. Isotropic hardening of the "compressive" yield surface forms the basis of this aspect of the model's inelastic response prediction when the principal stresses are dominantly compressive.

Calibration

A minimum of two experiments, uniaxial compression and uniaxial tension, is required to calibrate the simplest version of the concrete model (using all possible defaults and assuming temperature and field variable independence). Other experiments may be required to gain accuracy in postfailure behavior.

Uniaxial compression and tension tests

The uniaxial compression test involves compressing the sample between two rigid platens. The load and displacement in the direction of loading are recorded. From this, you can extract the stress-strain curve required for the concrete model directly. The uniaxial tension test is much more difficult to perform in the sense that it is necessary to have a stiff testing machine to be able to record the postfailure response. Quite often this test is not available, and you make an assumption about the tensile failure strength of the concrete (usually about 7%–10% of the compressive strength). The choice of tensile cracking stress is important; numerical problems may arise if very low cracking stresses are used (less than 1/100 or 1/1000 of the compressive strength).

Postcracking tensile behavior

The calibration of the postfailure response depends on the reinforcement present in the concrete. For plain concrete simulations the stress-displacement tension stiffening model should be used. Typical values for u_0 are 0.05 mm (2×10^{-3} in) for a normal concrete to 0.08 mm (3×10^{-3} in) for a high-strength concrete. For reinforced concrete simulations the stress-strain tension stiffening model should be used. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain of about 10 times the strain at failure. Since the strain at failure in standard concretes is typically 10^{-4}, this suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to a particular case.

Postcracking shear behavior

Combined tension and shear experiments are used to calibrate the postcracking shear behavior in Abaqus/Standard. These experiments are quite difficult to perform. If the test data are not available, a reasonable starting point is to assume that the shear retention factor, ϱ, goes linearly to zero at the same crack opening strain used for the tension stiffening model.

Biaxial yield and flow parameters

Biaxial experiments are required to calibrate the biaxial yield and flow parameters used to specify the failure ratios. If these are not available, the defaults can be used.

Temperature dependence

The calibration of temperature dependence requires the repetition of all the above experiments over the range of interest.

Comparison with experimental results

With proper calibration, the concrete model should produce reasonable results for mostly monotonic loadings. Comparison of the predictions of the model with the experimental results of Kupfer and Gerstle (1973) are shown in Figure 23.6.1–6 and Figure 23.6.1–7.

Elements

Abaqus/Standard offers a variety of elements for use with the smeared crack concrete model: beam, shell, plane stress, plane strain, generalized plane strain, axisymmetric, and three-dimensional elements.

For general shell analysis more than the default number of five integration points through the thickness of the shell should be used; nine thickness integration points are commonly used to model progressive failure of the concrete through the thickness with acceptable accuracy.

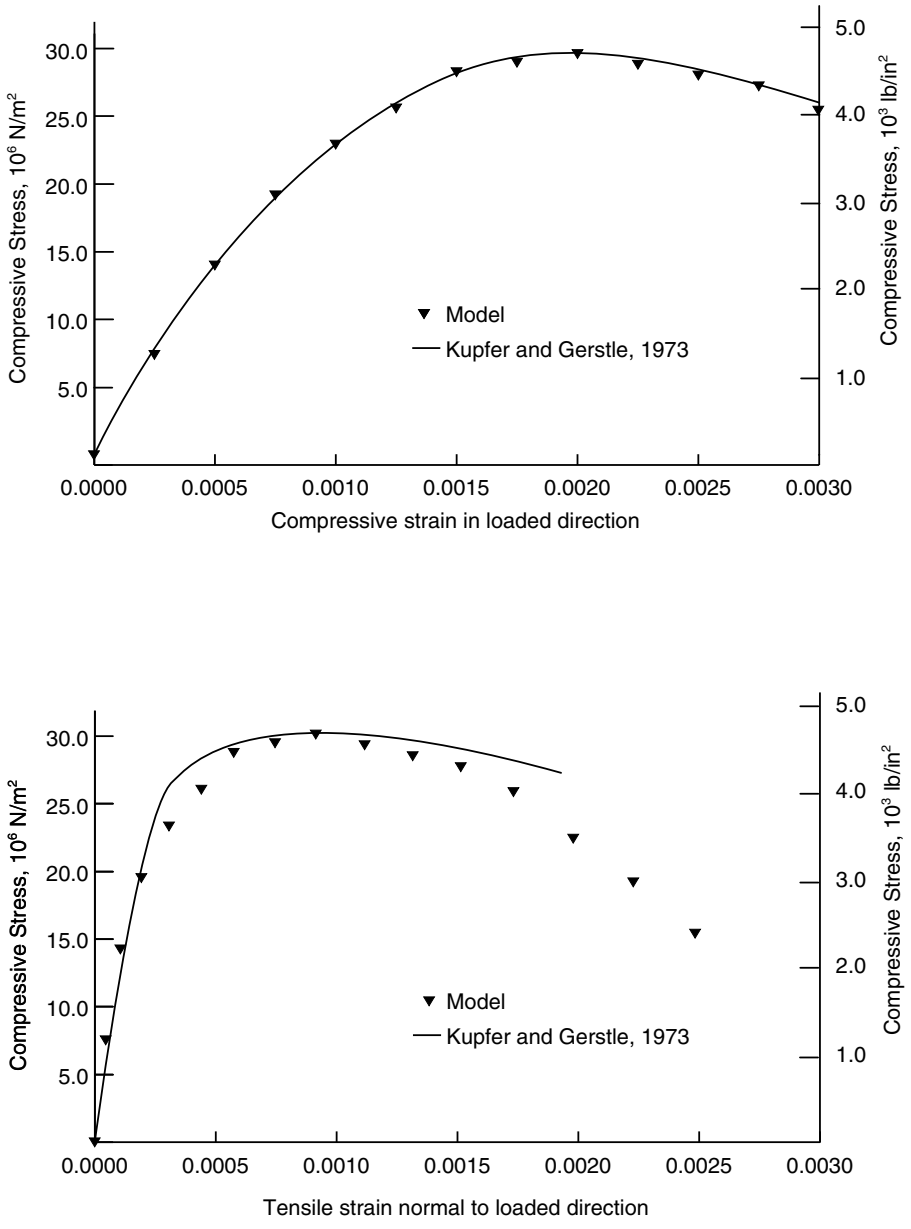


Figure 23.6.1–6 Comparison of model prediction and Kupfer and Gerstle's data for a uniaxial compression test.

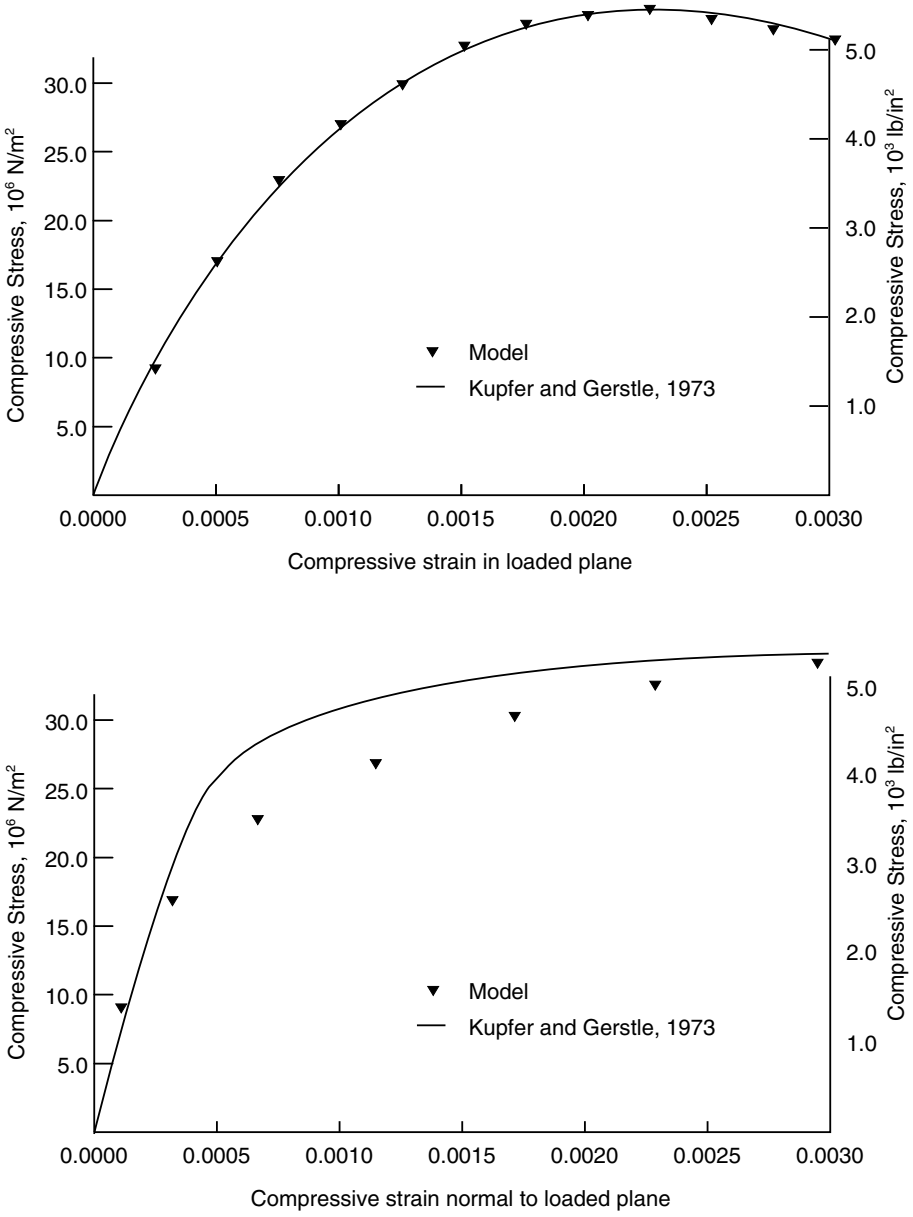


Figure 23.6.1–7 Comparison of model prediction and Kupfer and Gerstle's data for a biaxial compression test.

Output

In addition to the standard output identifiers available in Abaqus/Standard ("Abaqus/Standard output variable identifiers," Section 4.2.1), the following variables relate specifically to material points in the smeared crack concrete model:

CRACK	Unit normal to cracks in concrete.
CONF	Number of cracks at a concrete material point.

Additional references

- Crisfield, M. A., "Snap-Through and Snap-Back Response in Concrete Structures and the Dangers of Under-Integration," International Journal for Numerical Methods in Engineering, vol. 22, pp. 751–767, 1986.
- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," Cement and Concrete Research, vol. 6, pp. 773–782, 1976.
- Kupfer, H. B., and K. H. Gerstle, "Behavior of Concrete under Biaxial Stresses," Journal of Engineering Mechanics Division, ASCE, vol. 99, p. 853, 1973.

23.6.2 CRACKING MODEL FOR CONCRETE

Products: Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *BRITTLE CRACKING
- *BRITTLE FAILURE
- *BRITTLE SHEAR
- "Defining brittle cracking" in "Defining other mechanical models," Section 12.9.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The brittle cracking model in Abaqus/Explicit:

- provides a capability for modeling concrete in all types of structures: beams, trusses, shells and solids;
- can also be useful for modeling other materials such as ceramics or brittle rocks;
- is designed for applications in which the behavior is dominated by tensile cracking;
- assumes that the compressive behavior is always linear elastic;
- must be used with the linear elastic material model ("Linear elastic behavior," Section 22.2.1), which also defines the material behavior completely prior to cracking;
- is most accurate in applications where the brittle behavior dominates such that the assumption that the material is linear elastic in compression is adequate;
- can be used for plain concrete, even though it is intended primarily for the analysis of reinforced concrete structures;
- allows removal of elements based on a brittle failure criterion; and
- is defined in detail in "A cracking model for concrete and other brittle materials," Section 4.5.3 of the Abaqus Theory Manual.

See "Inelastic behavior," Section 23.1.1, for a discussion of the concrete models available in Abaqus.

Reinforcement

Reinforcement in concrete structures is typically provided by means of rebars. Rebars are one-dimensional strain theory elements (rods) that can be defined singly or embedded in oriented surfaces. Rebars are discussed in "Defining rebar as an element property," Section 2.2.4. They are typically used with elastic-plastic material behavior and are superposed on a mesh of standard element

types used to model the plain concrete. With this modeling approach, the concrete cracking behavior is considered independently of the rebar. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, are modeled approximately by introducing some "tension stiffening" into the concrete cracking model to simulate load transfer across cracks through the rebar.

Cracking

Abaqus/Explicit uses a smeared crack model to represent the discontinuous brittle behavior in concrete. It does not track individual "macro" cracks: instead, constitutive calculations are performed independently at each material point of the finite element model. The presence of cracks enters into these calculations by the way in which the cracks affect the stress and material stiffness associated with the material point.

For simplicity of discussion in this section, the term "crack" is used to mean a direction in which cracking has been detected at the single material calculation point in question: the closest physical concept is that there exists a continuum of micro-cracks in the neighborhood of the point, oriented as determined by the model. The anisotropy introduced by cracking is assumed to be important in the simulations for which the model is intended.

Crack directions

The Abaqus/Explicit cracking model assumes fixed, orthogonal cracks, with the maximum number of cracks at a material point limited by the number of direct stress components present at that material point of the finite element model (a maximum of three cracks in three-dimensional, plane strain, and axisymmetric problems; two cracks in plane stress and shell problems; and one crack in beam or truss problems). Internally, once cracks exist at a point, the component forms of all vector- and tensor-valued quantities are rotated so that they lie in the local system defined by the crack orientation vectors (the normals to the crack faces). The model ensures that these crack face normal vectors will be orthogonal, so that this local crack system is rectangular Cartesian. For output purposes you are offered results of stresses and strains in the global and/or local crack systems.

Crack detection

A simple Rankine criterion is used to detect crack initiation. This criterion states that a crack forms when the maximum principal tensile stress exceeds the tensile strength of the brittle material. Although crack detection is based purely on Mode I fracture considerations, ensuing cracked behavior includes both Mode I (tension softening/stiffening) and Mode II (shear softening/retention) behavior, as described later.

As soon as the Rankine criterion for crack formation has been met, we assume that a first crack has formed. The crack surface is taken to be normal to the direction of the maximum tensile principal stress. Subsequent cracks may form with crack surface normals in the direction of maximum principal tensile stress that is orthogonal to the directions of any existing crack surface normals at the same point.

Cracking is irrecoverable in the sense that, once a crack has occurred at a point, it remains throughout the rest of the calculation. However, crack closing and reopening may take place along the directions of the crack surface normals. The model neglects any permanent strain associated with cracking; that is, it is assumed that the cracks can close completely when the stress across them becomes compressive.

Tension stiffening

You can specify the postfailure behavior for direct straining across cracks by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion.

Postfailure stress-strain relation

In reinforced concrete the specification of postfailure behavior generally means giving the postfailure stress as a function of strain across the crack (Figure 23.6.2–1). In cases with little or no reinforcement, this introduces mesh sensitivity in the results, in the sense that the finite element predictions do not converge to a unique solution as the mesh is refined because mesh refinement leads to narrower crack bands.

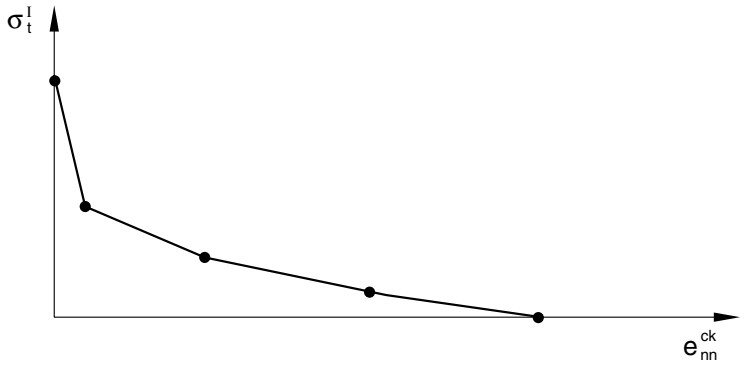


Figure 23.6.2–1 Postfailure stress-strain curve.

In practical calculations for reinforced concrete, the mesh is usually such that each element contains rebars. In this case the interaction between the rebars and the concrete tends to mitigate this effect, provided that a reasonable amount of "tension stiffening" is introduced in the cracking model to simulate this interaction. This requires an estimate of the tension stiffening effect, which depends on factors such as the density of reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the mesh. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain about ten times the strain at failure. Since the strain at failure in standard concretes is typically 10^{-4}, this suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to each particular case. In static applications too little tension stiffening will cause the local cracking failure in the concrete to introduce temporarily unstable behavior in the overall response of the model. Few practical designs exhibit such behavior, so that the presence of this type of response in the analysis model usually indicates that the tension stiffening is unreasonably low.

Input File Usage: *BRITTLE CRACKING, TYPE=STRAIN

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Brittle Cracking: **Type: Strain**

Fracture energy cracking criterion

When there is no reinforcement in significant regions of the model, the tension stiffening approach described above will introduce unreasonable mesh sensitivity into the results. However, it is generally accepted that Hillerborg's (1976) fracture energy proposal is adequate to allay the concern for many practical purposes. Hillerborg defines the energy required to open a unit area of crack in Mode I (G_f^I) as a material parameter, using brittle fracture concepts. With this approach the concrete's brittle behavior is characterized by a stress-*displacement* response rather than a stress-*strain* response. Under tension a concrete specimen will crack across some section; and its length, after it has been pulled apart sufficiently for most of the stress to be removed (so that the elastic strain is small), will be determined primarily by the opening at the crack, which does not depend on the specimen's length.

Implementation

In Abaqus/Explicit this fracture energy cracking model can be invoked by specifying the postfailure stress as a tabular function of displacement across the crack, as illustrated in Figure 23.6.2–2.

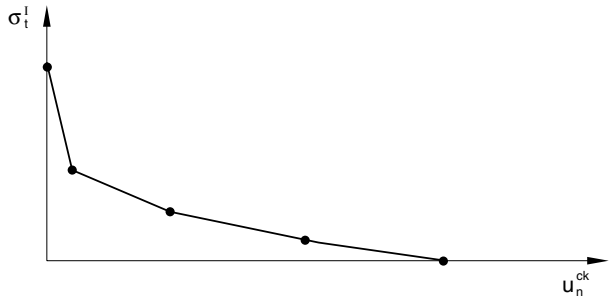


Figure 23.6.2–2 Postfailure stress-displacement curve.

Alternatively, the Mode I fracture energy, G_f^I, can be specified directly as a material property; in this case, define the failure stress, σ_{tu}^I, as a tabular function of the associated Mode I fracture energy. This model assumes a linear loss of strength after cracking (Figure 23.6.2–3). The crack normal displacement at which complete loss of strength takes place is, therefore, $u_{n0} = 2G_f^I/\sigma_{tu}^I$. Typical values of G_f^I range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high-strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2).

Input File Usage: Use the following option to specify the postfailure stress as a tabular function of displacement:

*BRITTLE CRACKING, TYPE=DISPLACEMENT

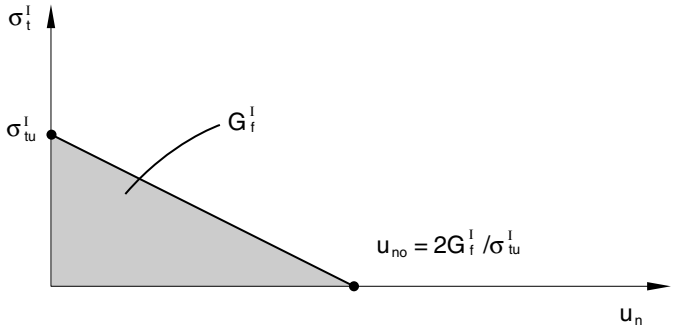


Figure 23.6.2–3 Postfailure stress-fracture energy curve.

Use the following option to specify the postfailure stress as a tabular function of the fracture energy:

*BRITTLE CRACKING, TYPE=GFI

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Brittle Cracking: **Type: Displacement** or **GFI**

Characteristic crack length

The implementation of the stress-displacement concept in a finite element model requires the definition of a characteristic length associated with a material point. The characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. We use this definition of the characteristic crack length because the direction in which cracks will occur is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect. Elements that are as close to square as possible are, therefore, recommended unless you can predict the direction in which cracks will form.

Shear retention model

An important feature of the cracking model is that, whereas crack initiation is based on Mode I fracture only, postcracked behavior includes Mode II as well as Mode I. The Mode II shear behavior is based on the common observation that the shear behavior depends on the amount of crack opening. More specifically, the cracked shear modulus is reduced as the crack opens. Therefore, Abaqus/Explicit offers a shear retention model in which the postcracked shear stiffness is defined as a function of the opening strain across the crack; the shear retention model must be defined in the cracking model, and zero shear retention should not be used.

In these models the dependence is defined by expressing the postcracking shear modulus, G_c, as a fraction of the uncracked shear modulus:

$$G_c = \rho(e_{nn}^{ck})\, G,$$

where G is the shear modulus of the uncracked material and the shear retention factor, $\rho(e_{nn}^{ck})$, depends on the crack opening strain, e_{nn}^{ck}. You can specify this dependence in piecewise linear form, as shown in Figure 23.6.2–4.

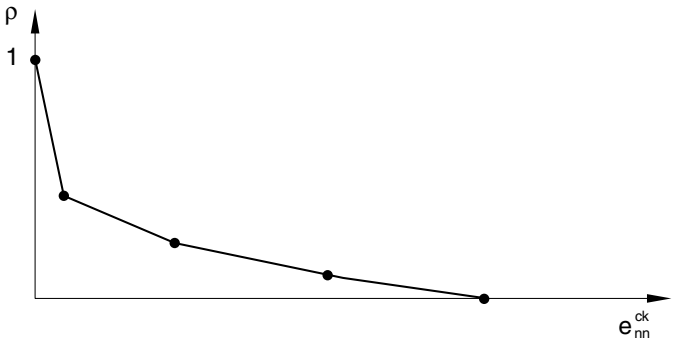


Figure 23.6.2–4 Piecewise linear form of the shear retention model.

Alternatively, shear retention can be defined in the power law form:

$$\rho(e_{nn}^{ck}) = \left(1 - \frac{e_{nn}^{ck}}{e_{max}^{ck}}\right)^{p},$$

where p and e_{max}^{ck} are material parameters. This form, shown in Figure 23.6.2–5, satisfies the requirements that $\rho \to 1$ as $e_{nn}^{ck} \to 0$ (corresponding to the state before crack initiation) and $\rho \to 0$ as $e_{nn}^{ck} \to e_{max}^{ck}$ (corresponding to complete loss of aggregate interlock). See "A cracking model for concrete and other brittle materials," Section 4.5.3 of the Abaqus Theory Manual, for a discussion of how shear retention is calculated in the case of two or more cracks.

Input File Usage: Use the following option to specify the piecewise linear form of the shear retention model:

*BRITTLE SHEAR, TYPE=RETENTION FACTOR

Use the following option to specify the power law form of the shear retention model:

*BRITTLE SHEAR, TYPE=POWER LAW

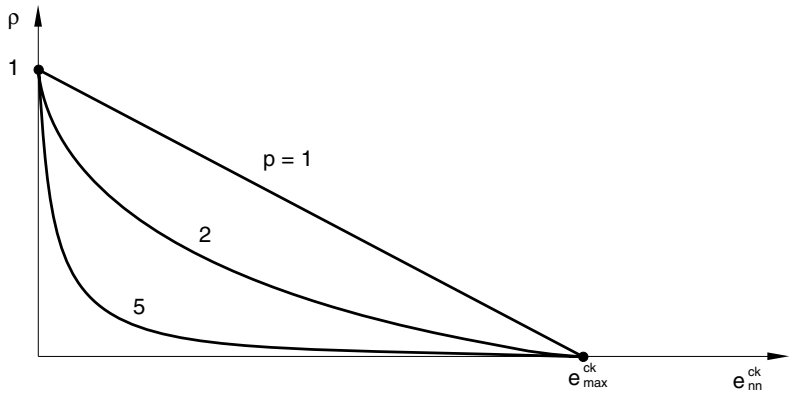


Figure 23.6.2–5 Power law form of the shear retention model.

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Brittle Cracking: **Suboptions→Brittle Shear**
Type: Retention Factor or **Power Law**

Calibration

One experiment, a uniaxial tension test, is required to calibrate the simplest version of the brittle cracking model. Other experiments may be required to gain accuracy in postfailure behavior.

Uniaxial tension test

This test is difficult to perform because it is necessary to have a very stiff testing machine to record the postcracking response. Quite often such equipment is not available; in this situation you must make an assumption about the tensile failure strength of the material and the postcracking response. For concrete the assumption usually made is that the tensile strength is 7–10% of the compressive strength. Uniaxial compression tests can be performed much more easily, so the compressive strength of concrete is usually known.

Postcracking tensile behavior

The values given for tension stiffening are a very important aspect of simulations using the Abaqus/Explicit brittle cracking model. The postcracking tensile response is highly dependent on the reinforcement present in the concrete. In simulations of unreinforced concrete, the tension stiffening models that are based on fracture energy concepts should be utilized. If reliable experimental data are not available, typical values that can be used were discussed before: common values of G_f^I range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high-strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2). In simulations of reinforced concrete the stress-strain tension stiffening model should be used; the amount of tension stiffening depends on the reinforcement present,

as discussed before. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain about ten times the strain at failure. Since the strain at failure in standard concretes is typically 10^{-4}, this suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to each particular case.

Postcracking shear behavior

Calibration of the postcracking shear behavior requires combined tension and shear experiments, which are difficult to perform. If such test data are not available, a reasonable starting point is to assume that the shear retention factor, ρ, goes linearly to zero at the same crack opening strain used for the tension stiffening model.

Brittle failure criterion

You can define brittle failure of the material. When one, two, or all three local direct cracking strain (displacement) components at a material point reach the value defined as the failure strain (displacement), the material point fails and all the stress components are set to zero. If all of the material points in an element fail, the element is removed from the mesh. For example, removal of a first-order reduced-integration solid element takes place as soon as its only integration point fails. However, all through-the-thickness integration points must fail before a shell element is removed from the mesh.

If the postfailure relation is defined in terms of stress versus strain, the failure strain must be given as the failure criterion. If the postfailure relation is defined in terms of stress versus displacement or stress versus fracture energy, the failure displacement must be given as the failure criterion. The failure strain (displacement) can be specified as a function of temperature and/or predefined field variables.

You can control how many cracks at a material point must fail before the material point is considered to have failed; the default is one crack. The number of cracks that must fail can only be one for beam and truss elements; it cannot be greater than two for plane stress and shell elements; and it cannot be greater than three otherwise.

Input File Usage: *BRITTLE FAILURE, CRACKS=*n*

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Brittle Cracking: **Suboptions→Brittle Failure** and select **Failure Criteria**: **Unidirectional**, **Bidirectional**, or **Tridirectional** to indicate the number of cracks that must fail for the material point to fail.

Determining when to use the brittle failure criterion

The brittle failure criterion is a crude way of modeling failure in Abaqus/Explicit and should be used with care. The main motivation for including this capability is to help in computations where not removing an element that can no longer carry stress may lead to excessive distortion of that element and subsequent premature termination of the simulation. For example, in a monotonically loaded structure whose failure mechanism is expected to be dominated by a single tensile macrofracture (Mode I cracking), it may be reasonable to use the brittle failure criterion to remove elements. On the other hand, the fact that the brittle material loses its ability to carry tensile stress does not preclude it from withstanding compressive stress;

therefore, it may not be appropriate to remove elements if the material is expected to carry compressive loads after it has failed in tension. An example may be a shear wall subjected to cyclic loading as a result of some earthquake excitation; in this case cracks that develop completely under tensile stress will be able to carry compressive stress when load reversal takes place.

Thus, the effective use of the brittle failure criterion relies on you having some knowledge of the structural behavior and potential failure mechanism. The use of the brittle failure criterion based on an incorrect user assumption of the failure mechanism will generally result in an incorrect simulation.

Selecting the number of cracks that must fail before the material point is considered to have failed

When you define brittle failure, you can control how many cracks must open to beyond the failure value before a material point is considered to have failed. The default number of cracks (one) should be used for most structural applications where failure is dominated by Mode I type cracking. However, there are cases in which you should specify a higher number because multiple cracks need to form to develop the eventual failure mechanism. One example may be an unreinforced, deep concrete beam where the failure mechanism is dominated by shear; in this case it is possible that two cracks need to form at each material point for the shear failure mechanism to develop.

Again, the appropriate choice of the number of cracks that must fail relies on your knowledge of the structural and failure behaviors.

Using brittle failure with rebar

It is possible to use the brittle failure criterion in brittle cracking elements for which rebar are also defined; the obvious application is the modeling of reinforced concrete. When such elements fail according to the brittle failure criterion, the brittle cracking contribution to the element stress carrying capacity is removed but the rebar contribution to the element stress carrying capacity is not removed. However, if you also include shear failure in the rebar material definition, the rebar contribution to the element stress carrying capacity will also be removed if the shear failure criterion specified for the rebar is satisfied. This allows the modeling of progressive failure of an under-reinforced concrete structure where the concrete fails first followed by ductile failure of the reinforcement.

Elements

Abaqus/Explicit offers a variety of elements for use with the cracking model: truss; shell; two-dimensional beam; and plane stress, plane strain, axisymmetric, and three-dimensional continuum elements. The model cannot be used with pipe and three-dimensional beam elements. Plane triangular, triangular prism, and tetrahedral elements are not recommended for use in reinforced concrete analysis since these elements do not support the use of rebar.

Output

In addition to the standard output identifiers available in Abaqus/Explicit (see “Abaqus/Explicit output variable identifiers,” Section 4.2.2), the following output variables relate directly to material points that use the brittle cracking model:

CKE	All cracking strain components.
CKLE	All cracking strain components in local crack axes.
CKEMAG	Cracking strain magnitude.
CKLS	All stress components in local crack axes.
CRACK	Crack orientations.
CKSTAT	Crack status of each crack.
STATUS	Status of element (brittle failure model). The status of an element is 1.0 if the element is active and 0.0 if the element is not.

Additional reference

- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," Cement and Concrete Research, vol. 6, pp. 773–782, 1976.

23.6.3 CONCRETE DAMAGED PLASTICITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Inelastic behavior," Section 23.1.1
- *CONCRETE DAMAGED PLASTICITY
- *CONCRETE TENSION STIFFENING
- *CONCRETE COMPRESSION HARDENING
- *CONCRETE TENSION DAMAGE
- *CONCRETE COMPRESSION DAMAGE
- "Defining concrete damaged plasticity" in "Defining plasticity," Section 12.9.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The concrete damaged plasticity model in Abaqus:

- provides a general capability for modeling concrete and other quasi-brittle materials in all types of structures (beams, trusses, shells, and solids);
- uses concepts of isotropic damaged elasticity in combination with isotropic tensile and compressive plasticity to represent the inelastic behavior of concrete;
- can be used for plain concrete, even though it is intended primarily for the analysis of reinforced concrete structures;
- can be used with rebar to model concrete reinforcement;
- is designed for applications in which concrete is subjected to monotonic, cyclic, and/or dynamic loading under low confining pressures;
- consists of the combination of nonassociated multi-hardening plasticity and scalar (isotropic) damaged elasticity to describe the irreversible damage that occurs during the fracturing process;
- allows user control of stiffness recovery effects during cyclic load reversals;
- can be defined to be sensitive to the rate of straining;
- can be used in conjunction with a viscoplastic regularization of the constitutive equations in Abaqus/Standard to improve the convergence rate in the softening regime;
- requires that the elastic behavior of the material be isotropic and linear (see "Defining isotropic elasticity" in "Linear elastic behavior," Section 22.2.1); and
- is defined in detail in "Damaged plasticity model for concrete and other quasi-brittle materials," Section 4.5.2 of the Abaqus Theory Manual.

See "Inelastic behavior," Section 23.1.1, for a discussion of the concrete models available in Abaqus.

Mechanical behavior

The model is a continuum, plasticity-based, damage model for concrete. It assumes that the main two failure mechanisms are tensile cracking and compressive crushing of the concrete material. The evolution of the yield (or failure) surface is controlled by two hardening variables, $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$, linked to failure mechanisms under tension and compression loading, respectively. We refer to $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ as tensile and compressive equivalent plastic strains, respectively. The following sections discuss the main assumptions about the mechanical behavior of concrete.

Uniaxial tension and compression stress behavior

The model assumes that the uniaxial tensile and compressive response of concrete is characterized by damaged plasticity, as shown in Figure 23.6.3–1. Under uniaxial tension the stress-strain response follows a linear elastic relationship until the value of the failure stress, σ_{t0}, is reached. The failure stress corresponds to the onset of micro-cracking in the concrete material. Beyond the failure stress the formation of micro-cracks is represented macroscopically with a softening stress-strain response, which induces strain localization in the concrete structure. Under uniaxial compression the response is linear until the value of initial yield, σ_{c0}. In the plastic regime the response is typically characterized by stress hardening followed by strain softening beyond the ultimate stress, σ_{cu}. This representation, although somewhat simplified, captures the main features of the response of concrete.

It is assumed that the uniaxial stress-strain curves can be converted into stress versus plastic-strain curves. (This conversion is performed automatically by Abaqus from the user-provided stress versus "inelastic" strain data, as explained below.) Thus,

$$\sigma_t = \sigma_t(\tilde{\varepsilon}_t^{pl}, \dot{\tilde{\varepsilon}}_t^{pl}, \theta, f_i),$$
$$\sigma_c = \sigma_c(\tilde{\varepsilon}_c^{pl}, \dot{\tilde{\varepsilon}}_c^{pl}, \theta, f_i),$$

where the subscripts *t* and *c* refer to tension and compression, respectively; $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ are the equivalent plastic strains, $\dot{\tilde{\varepsilon}}_t^{pl}$ and $\dot{\tilde{\varepsilon}}_c^{pl}$ are the equivalent plastic strain rates, θ is the temperature, and $f_i, (i = 1, 2, ...)$ are other predefined field variables.

As shown in Figure 23.6.3–1, when the concrete specimen is unloaded from any point on the strain softening branch of the stress-strain curves, the unloading response is weakened: the elastic stiffness of the material appears to be damaged (or degraded). The degradation of the elastic stiffness is characterized by two damage variables, d_t and d_c, which are assumed to be functions of the plastic strains, temperature, and field variables:

$$d_t = d_t(\tilde{\varepsilon}_t^{pl}, \theta, f_i); \quad 0 \le d_t \le 1,$$
$$d_c = d_c(\tilde{\varepsilon}_c^{pl}, \theta, f_i); \quad 0 \le d_c \le 1.$$

The damage variables can take values from zero, representing the undamaged material, to one, which represents total loss of strength.

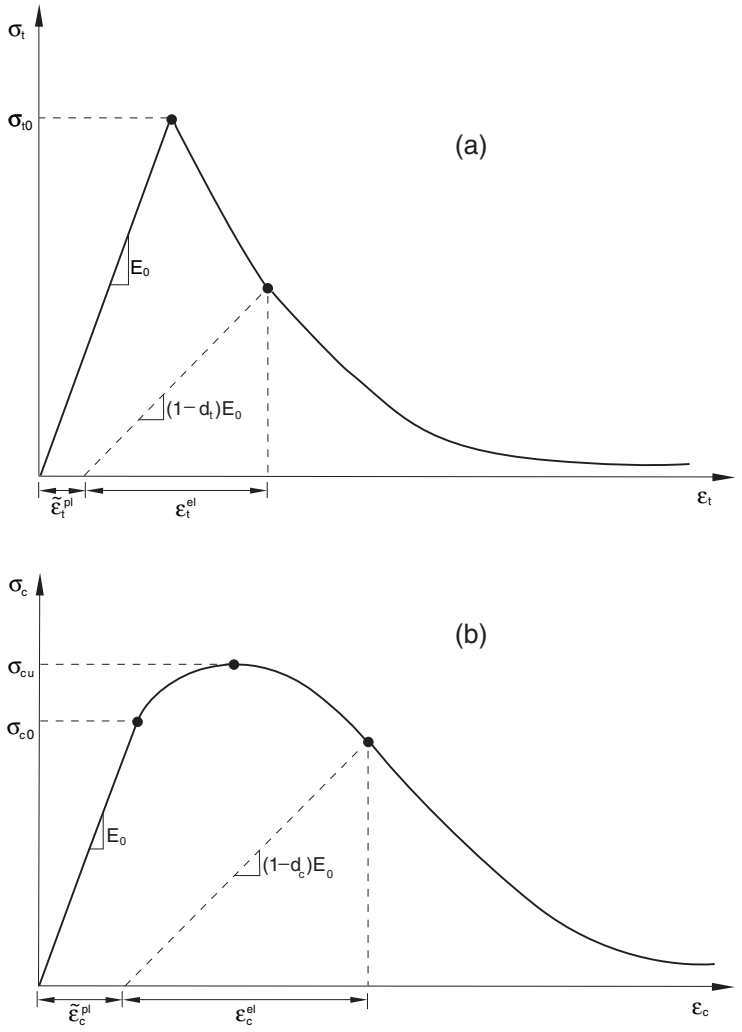


Figure 23.6.3–1 Response of concrete to uniaxial loading in tension (a) and compression (b).

If E_0 is the initial (undamaged) elastic stiffness of the material, the stress-strain relations under uniaxial tension and compression loading are, respectively:

$$\sigma_t = (1 - d_t)E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}),$$
$$\sigma_c = (1 - d_c)E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}).$$

We define the "effective" tensile and compressive cohesion stresses as

$$\bar{\sigma}_t = \frac{\sigma_t}{(1-d_t)} = E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}),$$
$$\bar{\sigma}_c = \frac{\sigma_c}{(1-d_c)} = E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}).$$

The effective cohesion stresses determine the size of the yield (or failure) surface.

Uniaxial cyclic behavior

Under uniaxial cyclic loading conditions the degradation mechanisms are quite complex, involving the opening and closing of previously formed micro-cracks, as well as their interaction. Experimentally, it is observed that there is some recovery of the elastic stiffness as the load changes sign during a uniaxial cyclic test. The stiffness recovery effect, also known as the "unilateral effect," is an important aspect of the concrete behavior under cyclic loading. The effect is usually more pronounced as the load changes from tension to compression, causing tensile cracks to close, which results in the recovery of the compressive stiffness.

The concrete damaged plasticity model assumes that the reduction of the elastic modulus is given in terms of a scalar degradation variable d as

$$E = (1-d)E_0,$$

where E_0 is the initial (undamaged) modulus of the material.

This expression holds both in the tensile ($\sigma_{11} > 0$) and the compressive ($\sigma_{11} < 0$) sides of the cycle. The stiffness degradation variable, d, is a function of the stress state and the uniaxial damage variables, d_t and d_c. For the uniaxial cyclic conditions Abaqus assumes that

$$(1-d) = (1-s_t d_c)(1-s_c d_t),$$

where s_t and s_c are functions of the stress state that are introduced to model stiffness recovery effects associated with stress reversals. They are defined according to

$$s_t = 1 - w_t r^*(\sigma_{11}); \quad 0 \le w_t \le 1,$$
$$s_c = 1 - w_c(1 - r^*(\sigma_{11})); \quad 0 \le w_c \le 1,$$

where

$$r^*(\sigma_{11}) = H(\sigma_{11}) = \begin{cases} 1 & \text{if} \quad \sigma_{11} > 0 \\ 0 & \text{if} \quad \sigma_{11} < 0 \end{cases}$$

The weight factors w_t and w_c, which are assumed to be material properties, control the recovery of the tensile and compressive stiffness upon load reversal. To illustrate this, consider the example in Figure 23.6.3–2, where the load changes from tension to compression. Assume that there was no previous compressive damage (crushing) in the material; that is, $\tilde{\varepsilon}_c^{pl} = 0$ and $d_c = 0$. Then

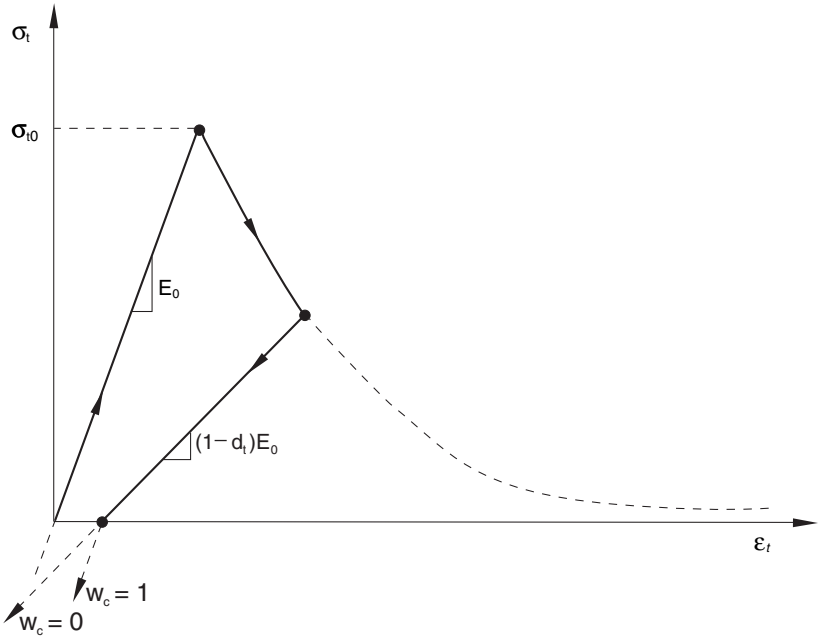


Figure 23.6.3–2 Illustration of the effect of the compression stiffness recovery parameter w_c.

$$(1-d) = (1-s_c d_t) = (1-(1-w_c(1-r^*))d_t).$$

- In tension ($\sigma_{11} > 0$), $r^* = 1$; therefore, $d = d_t$ as expected.
- In compression ($\sigma_{11} < 0$), $r^* = 0$, and $d = (1-w_c)d_t$. If $w_c = 1$, then $d = 0$; therefore, the material fully recovers the compressive stiffness (which in this case is the initial undamaged stiffness, $E = E_0$). If, on the other hand, $w_c = 0$, then $d = d_t$ and there is no stiffness recovery. Intermediate values of w_c result in partial recovery of the stiffness.

Multiaxial behavior

The stress-strain relations for the general three-dimensional multiaxial condition are given by the scalar damage elasticity equation:

$$\boldsymbol{\sigma} = (1-d)\mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}),$$

where $\mathbf{D}_0^{el}$ is the initial (undamaged) elasticity matrix.

The previous expression for the scalar stiffness degradation variable, d, is generalized to the multiaxial stress case by replacing the unit step function $r^*(\sigma_{11})$ with a multiaxial stress weight factor, $r(\hat{\boldsymbol{\sigma}})$, defined as

$$r(\hat{\boldsymbol{\sigma}}) = \frac{\sum_{i=1}^{3}\langle\hat{\sigma}_i\rangle}{\sum_{i=1}^{3}|\hat{\sigma}_i|}; \quad 0 \le r(\hat{\boldsymbol{\sigma}}) \le 1,$$

where $\hat{\sigma}_i$ $(i = 1, 2, 3)$ are the principal stress components. The Macauley bracket $\langle\cdot\rangle$ is defined by $\langle x\rangle = \frac{1}{2}(|x| + x)$.

See "Damaged plasticity model for concrete and other quasi-brittle materials," Section 4.5.2 of the Abaqus Theory Manual, for further details of the constitutive model.

Reinforcement

In Abaqus reinforcement in concrete structures is typically provided by means of rebars, which are one-dimensional rods that can be defined singly or embedded in oriented surfaces. Rebars are typically used with metal plasticity models to describe the behavior of the rebar material and are superposed on a mesh of standard element types used to model the concrete.

With this modeling approach, the concrete behavior is considered independently of the rebar. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, are modeled approximately by introducing some "tension stiffening" into the concrete modeling to simulate load transfer across cracks through the rebar. Details regarding tension stiffening are provided below.

Defining the rebar can be tedious in complex problems, but it is important that this be done accurately since it may cause an analysis to fail due to lack of reinforcement in key regions of a model. See "Defining rebar as an element property," Section 2.2.4, for more information regarding rebars.

Defining tension stiffening

The postfailure behavior for direct straining is modeled with tension stiffening, which allows you to define the strain-softening behavior for cracked concrete. This behavior also allows for the effects of the reinforcement interaction with concrete to be simulated in a simple manner. Tension stiffening is required in the concrete damaged plasticity model. You can specify tension stiffening by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion.

Postfailure stress-strain relation

In reinforced concrete the specification of postfailure behavior generally means giving the postfailure stress as a function of cracking strain, $\tilde{\varepsilon}_t^{ck}$. The cracking strain is defined as the total strain minus the elastic strain corresponding to the undamaged material; that is, $\tilde{\varepsilon}_t^{ck} = \varepsilon_t - \varepsilon_{0t}^{el}$, where $\varepsilon_{0t}^{el} = \sigma_t/E_0$, as illustrated in Figure 23.6.3–3. To avoid potential numerical problems, Abaqus enforces a lower limit on the postfailure stress equal to one-hundreth of the initial failure stress: $\sigma_t \ge \sigma_{t0}/100$.

Tension stiffening data are given in terms of the cracking strain, $\tilde{\varepsilon}_t^{ck}$. When unloading data are available, the data are provided to Abaqus in terms of tensile damage curves, $d_t - \tilde{\varepsilon}_t^{ck}$, as discussed below. Abaqus automatically converts the cracking strain values to plastic strain values using the relationship

$$\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck} - \frac{d_t}{(1-d_t)}\frac{\sigma_t}{E_0}.$$

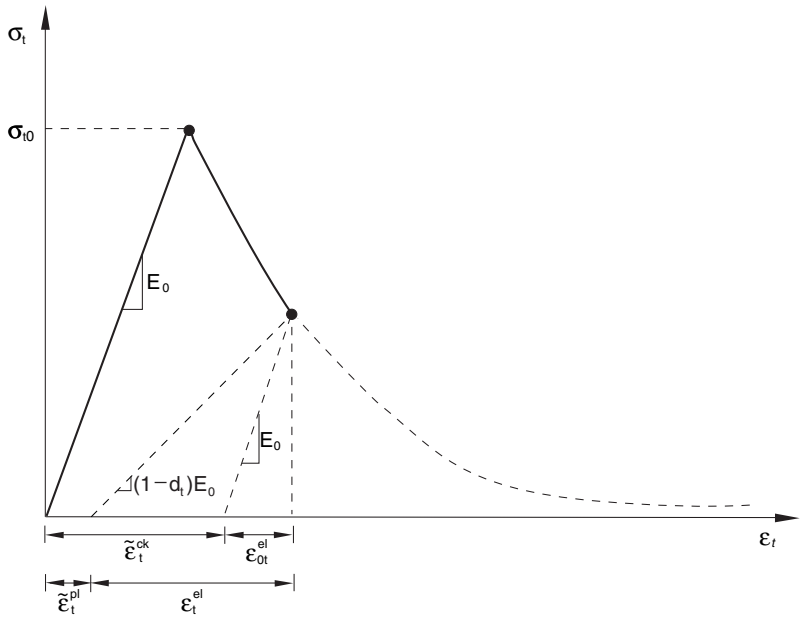


Figure 23.6.3–3 Illustration of the definition of the cracking strain $\tilde{\varepsilon}_t^{ck}$ used for the definition of tension stiffening data.

Abaqus will issue an error message if the calculated plastic strain values are negative and/or decreasing with increasing cracking strain, which typically indicates that the tensile damage curves are incorrect. In the absence of tensile damage $\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck}$.

In cases with little or no reinforcement, the specification of a postfailure stress-strain relation introduces mesh sensitivity in the results, in the sense that the finite element predictions do not converge to a unique solution as the mesh is refined because mesh refinement leads to narrower crack bands. This problem typically occurs if cracking failure occurs only at localized regions in the structure and mesh refinement does not result in the formation of additional cracks. If cracking failure is distributed evenly (either due to the effect of rebar or due to the presence of stabilizing elastic material, as in the case of plate bending), mesh sensitivity is less of a concern.

In practical calculations for reinforced concrete, the mesh is usually such that each element contains rebars. The interaction between the rebars and the concrete tends to reduce the mesh sensitivity, provided that a reasonable amount of tension stiffening is introduced in the concrete model to simulate this interaction. This requires an estimate of the tension stiffening effect, which depends on such factors as the density of reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the mesh. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that

the strain softening after failure reduces the stress linearly to zero at a total strain of about 10 times the strain at failure. The strain at failure in standard concretes is typically 10^{-4}, which suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to a particular case.

The choice of tension stiffening parameters is important since, generally, more tension stiffening makes it easier to obtain numerical solutions. Too little tension stiffening will cause the local cracking failure in the concrete to introduce temporarily unstable behavior in the overall response of the model. Few practical designs exhibit such behavior, so that the presence of this type of response in the analysis model usually indicates that the tension stiffening is unreasonably low.

Input File Usage: *CONCRETE TENSION STIFFENING, TYPE=STRAIN (default)

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**: **Tensile Behavior**: **Type: Strain**

Fracture energy cracking criterion

When there is no reinforcement in significant regions of the model, the tension stiffening approach described above will introduce unreasonable mesh sensitivity into the results. However, it is generally accepted that Hillerborg's (1976) fracture energy proposal is adequate to allay the concern for many practical purposes. Hillerborg defines the energy required to open a unit area of crack, G_f, as a material parameter, using brittle fracture concepts. With this approach the concrete's brittle behavior is characterized by a stress-displacement response rather than a stress-strain response. Under tension a concrete specimen will crack across some section. After it has been pulled apart sufficiently for most of the stress to be removed (so that the undamaged elastic strain is small), its length will be determined primarily by the opening at the crack. The opening does not depend on the specimen's length.

This fracture energy cracking model can be invoked by specifying the postfailure stress as a tabular function of cracking displacement, as shown in Figure 23.6.3–4.

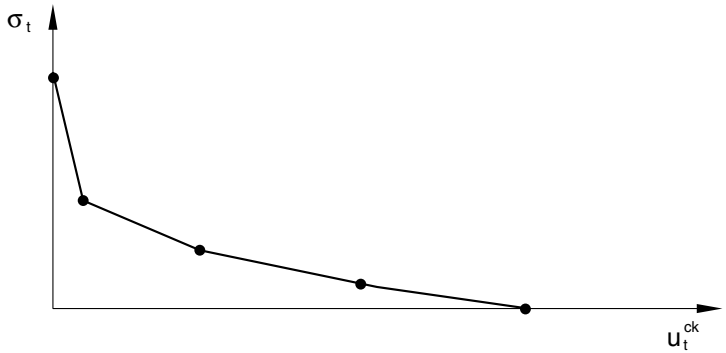


Figure 23.6.3–4 Postfailure stress-displacement curve.

Alternatively, the fracture energy, G_f, can be specified directly as a material property; in this case, define the failure stress, σ_{t0}, as a tabular function of the associated fracture energy. This model assumes a linear loss of strength after cracking, as shown in Figure 23.6.3–5.

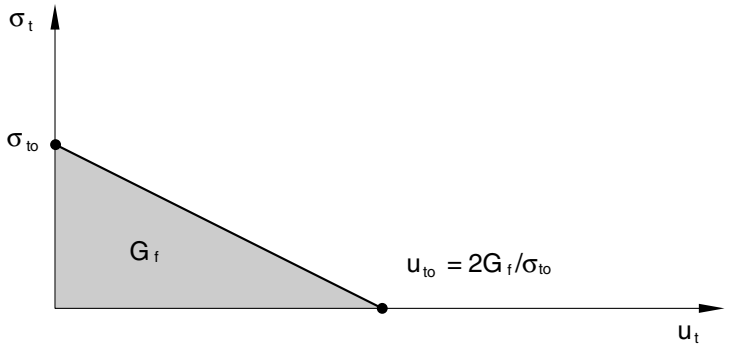


Figure 23.6.3–5 Postfailure stress-fracture energy curve.

The cracking displacement at which complete loss of strength takes place is, therefore, $u_{t0} = 2G_f/\sigma_{t0}$. Typical values of G_f range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high-strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2).

If tensile damage, d_t, is specified, Abaqus automatically converts the cracking displacement values to "plastic" displacement values using the relationship

$$u_t^{pl} = u_t^{ck} - \frac{d_t}{(1-d_t)}\frac{\sigma_t l_0}{E_0},$$

where the specimen length, l_0, is assumed to be one unit length, $l_0 = 1$.

Implementation

The implementation of this stress-displacement concept in a finite element model requires the definition of a characteristic length associated with an integration point. The characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic crack length is used because the direction in which cracking occurs is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect, and elements that have aspect ratios close to one are recommended.

Input File Usage: Use the following option to specify the postfailure stress as a tabular function of displacement:

*CONCRETE TENSION STIFFENING, TYPE=DISPLACEMENT

Use the following option to specify the postfailure stress as a tabular function of the fracture energy:

*CONCRETE TENSION STIFFENING, TYPE=GFI

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**: **Tensile Behavior**: **Type: Displacement** or **GFI**

Defining compressive behavior

You can define the stress-strain behavior of plain concrete in uniaxial compression outside the elastic range. Compressive stress data are provided as a tabular function of inelastic (or crushing) strain, $\tilde{\varepsilon}_c^{in}$, and, if desired, strain rate, temperature, and field variables. Positive (absolute) values should be given for the compressive stress and strain. The stress-strain curve can be defined beyond the ultimate stress, into the strain-softening regime.

Hardening data are given in terms of an inelastic strain, $\tilde{\varepsilon}_c^{in}$, instead of plastic strain, $\tilde{\varepsilon}_c^{pl}$. The compressive inelastic strain is defined as the total strain minus the elastic strain corresponding to the undamaged material, $\tilde{\varepsilon}_c^{in} = \varepsilon_c - \varepsilon_{0c}^{el}$, where $\varepsilon_{0c}^{el} = \sigma_c / E_0$, as illustrated in Figure 23.6.3–6. Unloading data are provided to Abaqus in terms of compressive damage curves, $d_c - \tilde{\varepsilon}_c^{in}$, as discussed below. Abaqus automatically converts the inelastic strain values to plastic strain values using the relationship

$$\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in} - \frac{d_c}{(1-d_c)} \frac{\sigma_c}{E_0}.$$

Abaqus will issue an error message if the calculated plastic strain values are negative and/or decreasing with increasing inelastic strain, which typically indicates that the compressive damage curves are incorrect. In the absence of compressive damage $\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in}$.

Input File Usage: *CONCRETE COMPRESSION HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**: **Compressive Behavior**

Defining damage and stiffness recovery

Damage, d_t and/or d_c, can be specified in tabular form. (If damage is not specified, the model behaves as a plasticity model; consequently, $\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck}$ and $\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in}$.)

In Abaqus the damage variables are treated as non-decreasing material point quantities. At any increment during the analysis, the new value of each damage variable is obtained as the maximum between the value at the end of the previous increment and the value corresponding to the current state (interpolated from the user-specified tabular data); that is,

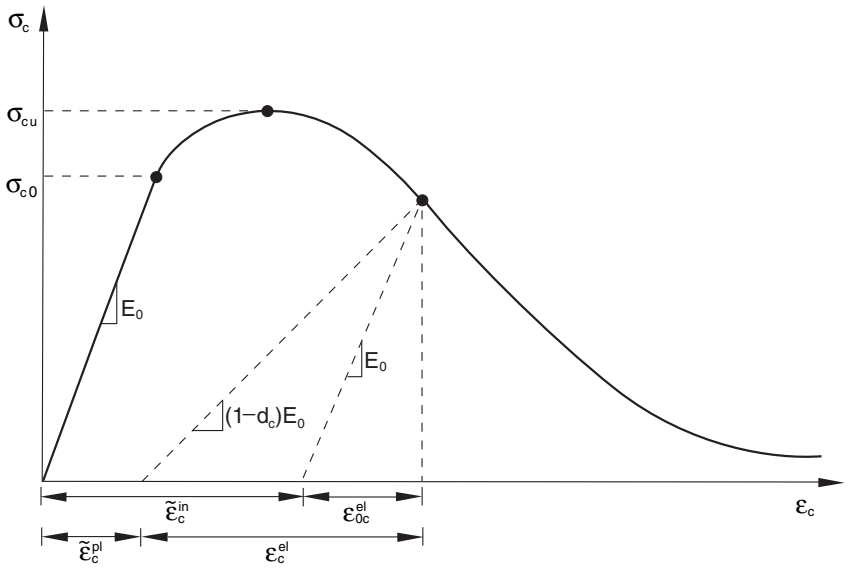


Figure 23.6.3–6 Definition of the compressive inelastic (or crushing) strain $\tilde{\varepsilon}_c^{in}$ used for the definition of compression hardening data.

$$d_t|_{t+\Delta t} = \max\left\{d_t|_t, d_t(\tilde{\varepsilon}_t^{pl}|_{t+\Delta t}, \theta|_{t+\Delta t}, f_i|_{t+\Delta t})\right\},$$
$$d_c|_{t+\Delta t} = \max\left\{d_c|_t, d_c(\tilde{\varepsilon}_c^{pl}|_{t+\Delta t}, \theta|_{t+\Delta t}, f_i|_{t+\Delta t})\right\}.$$

The choice of the damage properties is important since, generally, excessive damage may have a critical effect on the rate of convergence. It is recommended to avoid using values of the damage variables above 0.99, which corresponds to a 99% reduction of the stiffness.

Tensile damage

You can define the uniaxial tension damage variable, d_t, as a tabular function of either cracking strain or cracking displacement.

Input File Usage: Use the following option to specify tensile damage as a function of cracking strain:

∗CONCRETE TENSION DAMAGE, TYPE=STRAIN (default)

Use the following option to specify tensile damage as a function of cracking displacement:

∗CONCRETE TENSION DAMAGE, TYPE=DISPLACEMENT

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**: **Tensile Behavior**: **Suboptions→Tension Damage**: **Type: Strain** or **Displacement**

Compressive damage

You can define the uniaxial compression damage variable, d_c, as a tabular function of inelastic (crushing) strain.

Input File Usage: *CONCRETE COMPRESSION DAMAGE

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**: **Compressive Behavior**: **Suboptions→Compression Damage**

Stiffness recovery

As discussed above, stiffness recovery is an important aspect of the mechanical response of concrete under cyclic loading. Abaqus allows direct user specification of the stiffness recovery factors w_t and w_c.

The experimental observation in most quasi-brittle materials, including concrete, is that the compressive stiffness is recovered upon crack closure as the load changes from tension to compression. On the other hand, the tensile stiffness is not recovered as the load changes from compression to tension once crushing micro-cracks have developed. This behavior, which corresponds to $w_t = 0$ and $w_c = 1$, is the default used by Abaqus. Figure 23.6.3–7 illustrates a uniaxial load cycle assuming the default behavior.

Input File Usage: Use the following option to specify the compression stiffness recovery factor, w_c:

*CONCRETE TENSION DAMAGE, COMPRESSION RECOVERY=w_c

Use the following option to specify the tension stiffness recovery factor, w_t:

*CONCRETE COMPRESSION DAMAGE, TENSION RECOVERY=w_t

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**:
Tensile Behavior: **Suboptions→Tension Damage**: **Compression recovery:** w_c
Compressive Behavior: **Suboptions→Compression Damage**: **Tension recovery:** w_t

Rate dependence

The rate-sensitive behavior of quasi-brittle materials is mainly connected to the retardation effects that high strain rates have on the growth of micro-cracks. The effect is usually more pronounced under tensile loading. As the strain rate increases, the stress-strain curves exhibit decreasing nonlinearity as well as an increase in the peak strength. You can specify tension stiffening as a tabular function of cracking strain

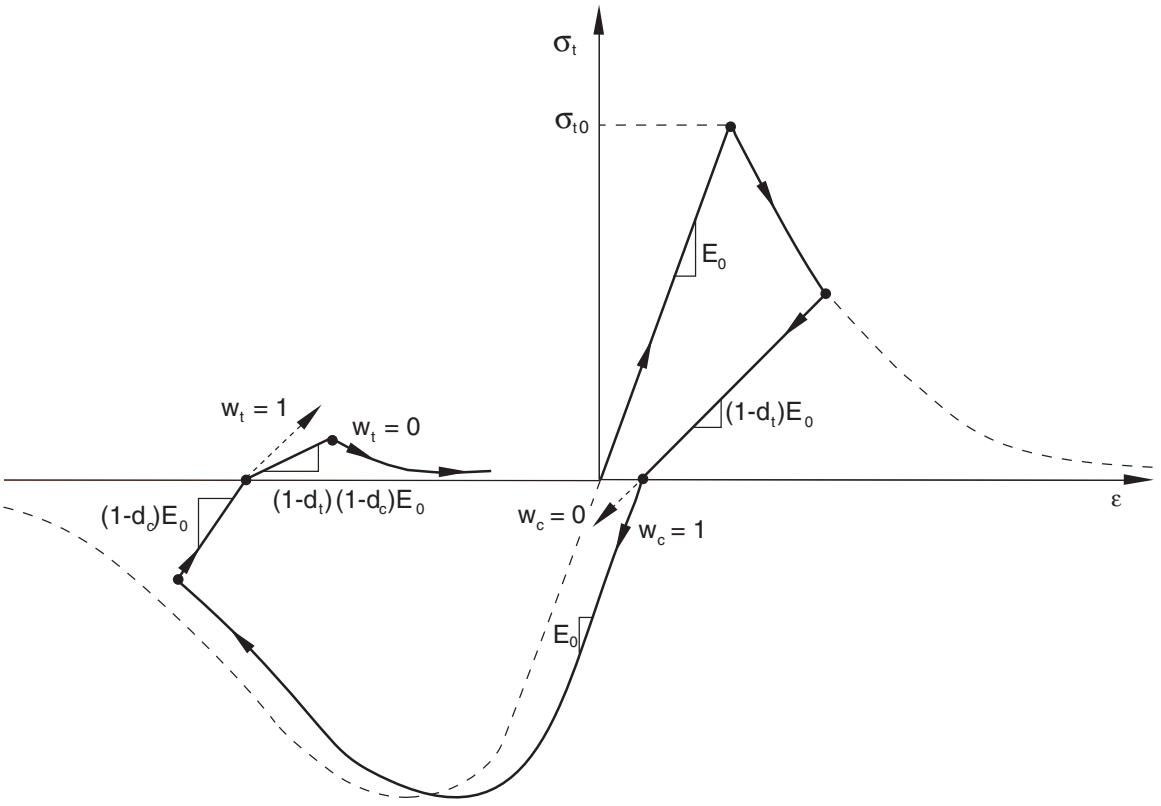


Figure 23.6.3–7 Uniaxial load cycle (tension-compression-tension) assuming default values for the stiffness recovery factors: $w_t = 0$ and $w_c = 1$.

(or displacement) rate, and you can specify compression hardening data as a tabular function of inelastic strain rate.

Input File Usage: Use the following options:

*CONCRETE TENSION STIFFENING
*CONCRETE COMPRESSION HARDENING

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**:
Tensile Behavior: **Use strain-rate-dependent data**
Compressive Behavior: **Use strain-rate-dependent data**

Concrete plasticity

You can define flow potential, yield surface, and in Abaqus/Standard viscosity parameters for the concrete damaged plasticity material model.

Input File Usage: *CONCRETE DAMAGED PLASTICITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Concrete Damaged Plasticity**: **Plasticity**

Effective stress invariants

The effective stress is defined as

$$\bar{\boldsymbol{\sigma}} = \mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}).$$

The plastic flow potential function and the yield surface make use of two stress invariants of the effective stress tensor, namely the hydrostatic pressure stress,

$$\bar{p} = -\frac{1}{3}\mathrm{trace}\,(\bar{\boldsymbol{\sigma}}),$$

and the Mises equivalent effective stress,

$$\bar{q} = \sqrt{\frac{3}{2}(\bar{\mathbf{S}} : \bar{\mathbf{S}})},$$

where $\bar{\mathbf{S}}$ is the effective stress deviator, defined as

$$\bar{\mathbf{S}} = \bar{\boldsymbol{\sigma}} + \bar{p}\mathbf{I}.$$

Plastic flow

The concrete damaged plasticity model assumes nonassociated potential plastic flow. The flow potential G used for this model is the Drucker-Prager hyperbolic function:

$$G = \sqrt{(\epsilon\sigma_{t0}\tan\psi)^2 + \bar{q}^2} - \bar{p}\tan\psi,$$

where

$\psi(\theta, f_i)$ is the dilation angle measured in the p–q plane at high confining pressure;

$\sigma_{t0}(\theta, f_i) = \sigma_t|_{\tilde{\varepsilon}_t^{pl}=0, \dot{\tilde{\varepsilon}}_t^{pl}=0}$ is the uniaxial tensile stress at failure, taken from the user-specified tension stiffening data; and

$\epsilon(\theta, f_i)$ is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as the eccentricity tends to zero).

This flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches the linear Drucker-Prager flow potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. See "Models for granular or polymer behavior," Section 4.4.2 of the Abaqus Theory Manual, for further discussion of this potential.

The default flow potential eccentricity is $\epsilon = 0.1$, which implies that the material has almost the same dilation angle over a wide range of confining pressure stress values. Increasing the value of ϵ provides more curvature to the flow potential, implying that the dilation angle increases more rapidly as the confining pressure decreases. Values of ϵ that are significantly less than the default value may lead to convergence problems if the material is subjected to low confining pressures because of the very tight curvature of the flow potential locally where it intersects the $\boldsymbol{p}$-axis.

Yield function

The model makes use of the yield function of Lubliner et. al. (1989), with the modifications proposed by Lee and Fenves (1998) to account for different evolution of strength under tension and compression. The evolution of the yield surface is controlled by the hardening variables, $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$. In terms of effective stresses, the yield function takes the form

$$F = \frac{1}{1-\alpha}\left(\bar{q} - 3\alpha\bar{p} + \beta(\tilde{\boldsymbol{\varepsilon}}^{pl})\langle\hat{\bar{\sigma}}_{\max}\rangle - \gamma\langle-\hat{\bar{\sigma}}_{\max}\rangle\right) - \bar{\sigma}_c(\tilde{\varepsilon}_c^{pl}) = 0,$$

with

$$\alpha = \frac{(\sigma_{b0}/\sigma_{c0}) - 1}{2(\sigma_{b0}/\sigma_{c0}) - 1};\ 0 \le \alpha \le 0.5,$$
$$\beta = \frac{\bar{\sigma}_c(\tilde{\varepsilon}_c^{pl})}{\bar{\sigma}_t(\tilde{\varepsilon}_t^{pl})}(1-\alpha) - (1+\alpha),$$
$$\gamma = \frac{3(1-K_c)}{2K_c - 1}.$$

Here,

$\hat{\bar{\sigma}}_{\max}$	is the maximum principal effective stress;
σ_{b0}/σ_{c0}	is the ratio of initial equibiaxial compressive yield stress to initial uniaxial compressive yield stress (the default value is 1.16);
K_c	is the ratio of the second stress invariant on the tensile meridian, $q_{(\mathrm{TM})}$, to that on the compressive meridian, $q_{(\mathrm{CM})}$, at initial yield for any given value of the pressure invariant $\boldsymbol{p}$ such that the maximum principal stress is negative, $\hat{\sigma}_{\max} < 0$ (see Figure 23.6.3–8); it must satisfy the condition $0.5 < K_c \le 1.0$ (the default value is $2/3$);
$\bar{\sigma}_t(\tilde{\varepsilon}_t^{pl})$	is the effective tensile cohesion stress; and
$\bar{\sigma}_c(\tilde{\varepsilon}_c^{pl})$	is the effective compressive cohesion stress.

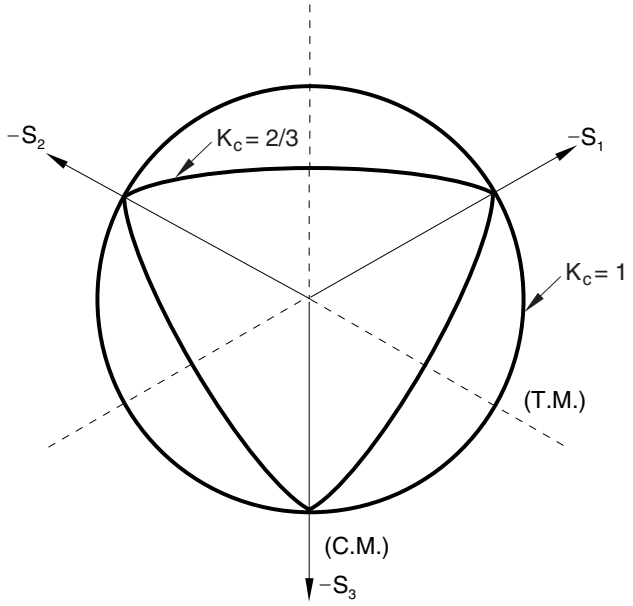


Figure 23.6.3–8 Yield surfaces in the deviatoric plane, corresponding to different values of K_c.

Typical yield surfaces are shown in Figure 23.6.3–8 on the deviatoric plane and in Figure 23.6.3–9 for plane stress conditions.

Nonassociated flow

Because plastic flow is nonassociated, the use of concrete damaged plasticity results in a nonsymmetric material stiffness matrix. Therefore, to obtain an acceptable rate of convergence in Abaqus/Standard, the unsymmetric matrix storage and solution scheme should be used. Abaqus/Standard will automatically activate the unsymmetric solution scheme if concrete damaged plasticity is used in the analysis. If desired, you can turn off the unsymmetric solution scheme for a particular step (see "Defining an analysis," Section 6.1.2).

Viscoplastic regularization

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. A common technique to overcome some of these convergence difficulties is the use of a viscoplastic regularization of the constitutive equations, which causes the consistent tangent stiffness of the softening material to become positive for sufficiently small time increments.

The concrete damaged plasticity model can be regularized in Abaqus/Standard using viscoplasticity by permitting stresses to be outside of the yield surface. We use a generalization of the Duvaut-Lions regularization, according to which the viscoplastic strain rate tensor, $\dot{\varepsilon}_v^{pl}$, is defined as

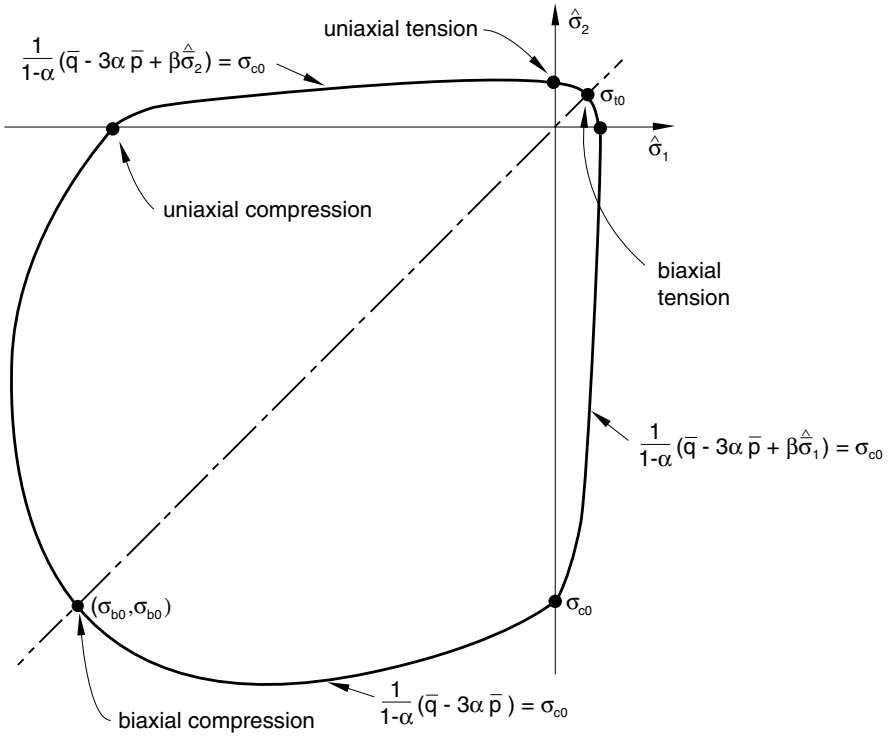


Figure 23.6.3–9 Yield surface in plane stress.

$$\dot{\varepsilon}_v^{pl} = \frac{1}{\mu}(\varepsilon^{pl} - \varepsilon_v^{pl}).$$

Here μ is the viscosity parameter representing the relaxation time of the viscoplastic system, and ε^{pl} is the plastic strain evaluated in the inviscid backbone model.

Similarly, a viscous stiffness degradation variable, d_v, for the viscoplastic system is defined as

$$\dot{d}_v = \frac{1}{\mu}(d - d_v),$$

where d is the degradation variable evaluated in the inviscid backbone model. The stress-strain relation of the viscoplastic model is given as

$$\boldsymbol{\sigma} = (1 - d_v)\mathbf{D}_0^{el} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_v^{pl}).$$

Using the viscoplastic regularization with a small value for the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the

model in the softening regime, without compromising results. The basic idea is that the solution of the viscoplastic system relaxes to that of the inviscid case as $t/\mu \to \infty$, where t represents time. You can specify the value of the viscosity parameter as part of the concrete damaged plasticity material behavior definition. If the viscosity parameter is different from zero, output results of the plastic strain and stiffness degradation refer to the viscoplastic values, $\boldsymbol{\varepsilon}_v^{pl}$ and d_v. In Abaqus/Standard the default value of the viscosity parameter is zero, so that no viscoplastic regularization is performed.

Material damping

The concrete damaged plasticity model can be used in combination with material damping (see "Material damping," Section 26.1.1). If stiffness proportional damping is specified, Abaqus calculates the damping stress based on the undamaged elastic stiffness. This may introduce large artificial damping forces on elements undergoing severe damage at high strain rates.

Visualization of "crack directions"

Unlike concrete models based on the smeared crack approach, the concrete damaged plasticity model does not have the notion of cracks developing at the material integration point. However, it is possible to introduce the concept of an effective crack direction with the purpose of obtaining a graphical visualization of the cracking patterns in the concrete structure. Different criteria can be adopted within the framework of scalar-damage plasticity for the definition of the direction of cracking. Following Lubliner et. al. (1989), we can assume that cracking initiates at points where the tensile equivalent plastic strain is greater than zero, $\tilde{\varepsilon}_t^{pl} > 0$, and the maximum principal plastic strain is positive. The direction of the vector normal to the crack plane is assumed to be parallel to the direction of the maximum principal plastic strain. This direction can be viewed in the Visualization module of Abaqus/CAE.

Abaqus/CAE Usage: Visualization module:
Result→Field Output: **PE**, **Max. Principal**
Plot→Symbols

Elements

Abaqus offers a variety of elements for use with the concrete damaged plasticity model: truss, shell, plane stress, plane strain, generalized plane strain, axisymmetric, and three-dimensional elements. Most beam elements can be used; however, beam elements in space that include shear stress caused by torsion and do not include hoop stress (such as B31, B31H, B32, B32H, B33, and B33H) cannot be used. Thin-walled, open-section beam elements and PIPE elements can be used with the concrete damaged plasticity model in Abaqus/Standard.

For general shell analysis more than the default number of five integration points through the thickness of the shell should be used; nine thickness integration points are commonly used to model progressive failure of the concrete through the thickness with acceptable accuracy.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables relate specifically to material points in the concrete damaged plasticity model:

DAMAGEC	Compressive damage variable, d_c.
DAMAGET	Tensile damage variable, d_t.
PEEQ	Compressive equivalent plastic strain, $\tilde{\varepsilon}_c^{pl}$.
PEEQT	Tensile equivalent plastic strain, $\tilde{\varepsilon}_t^{pl}$.
SDEG	Stiffness degradation variable, d.
DMENER	Energy dissipated per unit volume by damage.
ELDMD	Total energy dissipated in the element by damage.
ALLDMD	Energy dissipated in the whole (or partial) model by damage. The contribution from ALLDMD is included in the total strain energy ALLIE.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
SENER	The recoverable part of the energy per unit volume.
ELSE	The recoverable part of the energy in the element.
ALLSE	The recoverable part of the energy in the whole (partial) model.
ESEDEN	The recoverable part of the energy per unit volume in the element.

Additional references

- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," Cement and Concrete Research, vol. 6, pp. 773–782, 1976.
- Lee, J., and G. L. Fenves, "Plastic-Damage Model for Cyclic Loading of Concrete Structures," Journal of Engineering Mechanics, vol. 124, no. 8, pp. 892–900, 1998.
- Lubliner, J., J. Oliver, S. Oller, and E. Oñate, "A Plastic-Damage Model for Concrete," International Journal of Solids and Structures, vol. 25, pp. 299–329, 1989.

23.7 Permanent set in rubberlike materials

- "Permanent set in rubberlike materials," Section 23.7.1

23.7.1 PERMANENT SET IN RUBBERLIKE MATERIALS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Combining material behaviors," Section 21.1.3
- "Hyperelastic behavior of rubberlike materials," Section 22.5.1
- "Classical metal plasticity," Section 23.2.1
- *HYPERELASTIC
- *MULLINS EFFECT
- *PLASTIC

Overview

This feature:

- is intended for modeling permanent set observed in filled elastomers and thermoplastics;
- is based on multiplicative split of the deformation gradient;
- is based on the theory of incompressible isotropic hardening plasticity;
- can be used with any isotropic hyperelasticity model;
- can be combined with Mullins effects; and
- cannot be used to model viscoelastic or hysteresis effects or with the steady-state transport procedure.

Material behavior

The real behavior of filled rubber elastomers under cyclic loading conditions is quite complex as shown in Figure 23.7.1–1. The observed mechanical behaviors are progressive damage resulting in a reduction of load carrying capacity with each cycle, stress softening (also known as Mullins effect) upon reloading after the first unloading from a previously attained maximum strain level, hysteretic dissipation of energy, and permanent set. This section is concerned with modeling permanent set; therefore, the idealized representation of permanent set is described below.

Idealized material behavior

From Figure 23.7.1–1 it is clear that the observed permanent set is different for each cycle, but the material has a tendency to stabilize after a number of cycles of loading between zero stress and a given level of strain. For a given load level along the primary loading path shown with the dashed line in Figure 23.7.1–1, the idealized representation of permanent set will be a single strain value after unloading has taken place. Since rate and time effects are ignored in this model, idealized loading and unloading take place along the same path, whether Mullins effect is included or not.

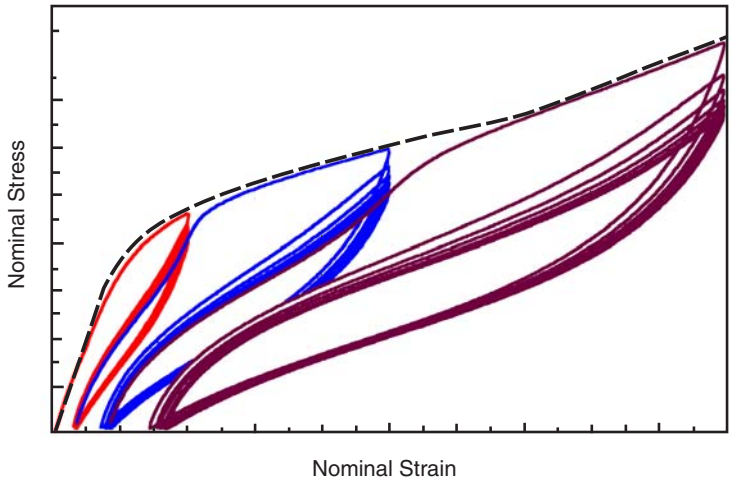


Figure 23.7.1–1 Typical behavior of a filled elastomer.

The permanent set behavior is captured by isotropic hardening Mises plasticity with an associated flow rule. In the context of finite elastic strains associated with the underlying rubberlike material, plasticity is modeled using a multiplicative split of the deformation gradient into elastic and plastic components:

$$\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p,$$

where $\mathbf{F}^e$ is the elastic part of the deformation gradient (representing the hyperelastic behavior) and $\mathbf{F}^p$ is the plastic part of the deformation gradient (representing the stress-free intermediate configuration).

An example of modeling permanent set along with Mullins effect for a rubberlike material can be found in "Analysis of a solid disc with Mullins effect and permanent set," Section 3.1.7 of the Abaqus Example Problems Manual.

Specifying permanent set

The primary hyperelastic behavior can be defined by using any of the hyperelastic material models (see "Hyperelastic behavior of rubberlike materials," Section 22.5.1). If test data input is used to define the hyperelastic response of the material, the data must be specified with respect to the stress-free intermediate configuration after unloading has taken place.

Permanent set can be defined through an isotropic hardening function in terms of the yield stress and the equivalent plastic strain. In this case the yield stress is the (effective) Kirchoff stress on the primary loading path from which unloading takes place, and the equivalent plastic strain is the corresponding logarithmic permanent set observed in the material. If σ is the true (Cauchy) stress, Kirchoff stress is defined as $J\sigma$, where J is the determinant of $\mathbf{F}$.

Depending on what is being modeled, permanent set may be defined as the true permanent set seen in the material after recovery of viscoelastic strains or it may include viscoelastic strains. In either case, an initial yield stress is required, below which there will be no permanent set and the behavior of the material will be fully elastic. In the case of filled rubbers this initial yield stress may correspond to a small nonzero stress; whereas for the family of thermoplastic materials, there may be a more marked value of initial yield stress.

Input File Usage: *PLASTIC, HARDENING=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Plasticity→Plastic**

Processing test data

If you have uniaxial and/or biaxial test data, as shown in Figure 23.7.1–1, you can use an interactive Abaqus/CAE plug-in to obtain the hyperelasticity, plasticity, and Mullins effect data. For information about the plug-in and instructions about its usage, see "Abaqus/CAE plug-in application for processing cyclic test data of filled elastomers and thermoplastics" in the Dassault Systèmes Knowledge Base at www.3ds.com/support/knowledge-base or the SIMULIA Online Support System, which is accessible through the **My Support** page at www.simulia.com.

Limitations

The model is intended to capture permanent set under multiaxial stress states and mild reverse loading conditions, as illustrated by Govindarajan, Hurtado, and Mars (2007). This model is not intended to capture deformation under complete reverse loading. Any rate effects apply only to the plastic part of the material definition.

Elements

Permanent set can be modeled with all element types that support the use of the hyperelastic material model.

Procedures

Permanent set modeling can be carried out in all procedures that support the use of the hyperelastic material model with the exception of the steady-state transport procedure. In linear perturbation steps in Abaqus/Standard, the current material tangent stiffness corresponding to the elastic part is used to determine the response, while ignoring any plasticity effects.

Output

The standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2) corresponding to other isotropic hardening plasticity models can be obtained for permanent set models.

Additional references

- Govindarajan, S. M., J. A. Hurtado, and W. V. Mars, "Simulation of Mullins Effect in Filled Elastomers Using Multiplicative Decomposition," European Conference for Constitutive Models for Rubber, September 2007, Paris, France.
- Simo, J. C., "Algorithms for Static and Dynamic Multiplicative Plasticity that Preserve the Classical Return Mapping Schemes of the Infinitesimal Theory," Computer Methods in Applied Mechanics and Engineering, vol. 99, p. 61–112, 1992.
- Weber, G., and L. Anand, "Finite Deformation Constitutive Equations and Time Integration Procedure for Isotropic Hyperelastic-Viscoplastic Solids," Computer Methods in Applied Mechanics and Engineering, vol. 79, p. 173–202, 1990.

24. Progressive Damage and Failure

Progressive damage and failure: overview 24.1
Damage and failure for ductile metals 24.2
Damage and failure for fiber-reinforced composites 24.3
Damage and failure for ductile materials in low-cycle fatigue analysis 24.4

24.1 Progressive damage and failure: overview

- "Progressive damage and failure," Section 24.1.1

24.1.1 PROGRESSIVE DAMAGE AND FAILURE

Abaqus provides the following models to predict progressive damage and failure:

- **Progressive damage and failure for ductile metals:** Abaqus offers a general capability for modeling progressive damage and failure in ductile metals. The functionality can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models ("Damage and failure for ductile metals: overview," Section 24.2.1). The capability supports the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), Müschenborn-Sonne forming limit diagram (MSFLD), and Marciniak-Kuczynski (M-K) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The progressive damage models allow for a smooth degradation of the material stiffness, which makes them suitable for both quasi-static and dynamic situations, a great advantage over the dynamic failure models ("Dynamic failure models," Section 23.2.8).

  The Johnson-Cook and Marciniak-Kuczynski (M-K) damage initiation criteria are not available in Abaqus/Standard.

- **Progressive damage and failure for fiber-reinforced materials:** Abaqus offers a capability to model anisotropic damage in fiber-reinforced materials ("Damage and failure for fiber-reinforced composites: overview," Section 24.3.1). The response of the undamaged material is assumed to be linearly elastic, and the model is intended to predict behavior of fiber-reinforced materials for which damage can be initiated without a large amount of plastic deformation. The Hashin's initiation criteria are used to predict the onset of damage, and the damage evolution law is based on the energy dissipated during the damage process and linear material softening.

- **Progressive damage and failure for ductile materials in low-cycle fatigue analysis:** Abaqus/Standard offers a capability to model progressive damage and failure for ductile materials due to stress reversals and the accumulation of inelastic strain in a low-cycle fatigue analysis using the direct cyclic approach (see "Low-cycle fatigue analysis using the direct cyclic approach," Section 6.2.7). The damage initiation criterion and damage evolution are characterized by the accumulated inelastic hysteresis energy per stabilized cycle (see "Damage and failure for ductile materials in low-cycle fatigue analysis: overview," Section 24.4.1). After damage initiation, the elastic material stiffness is degraded progressively according to the specified damage evolution response.

In addition, Abaqus offers a concrete damaged model ("Concrete damaged plasticity," Section 23.6.3), dynamic failure models ("Dynamic failure models," Section 23.2.8), and specialized capabilities for modeling damage and failure in cohesive elements ("Defining the constitutive response of cohesive elements using a traction-separation description," Section 32.5.6) and in connectors ("Connector damage behavior," Section 31.2.7).

This section provides an overview of the progressive damage and failure capability and a brief description of the concepts of damage initiation and evolution. The discussion in this section is limited to damage models for ductile metals and fiber-reinforced materials.

General framework for modeling damage and failure

Abaqus offers a general framework for material failure modeling that allows the combination of multiple failure mechanisms acting simultaneously on the same material. Material failure refers to the complete loss of load-carrying capacity that results from progressive degradation of the material stiffness. The stiffness degradation process is modeled using damage mechanics.

To help understand the failure modeling capabilities in Abaqus, consider the response of a typical metal specimen during a simple tensile test. The stress-strain response, such as that illustrated in Figure 24.1.1–1, will show distinct phases. The material response is initially linear elastic, $a - b$, followed by plastic yielding with strain hardening, $b - c$. Beyond point ***c*** there is a marked reduction of load-carrying capacity until rupture, $c - d$. The deformation during this last phase is localized in a neck region of the specimen. Point ***c*** identifies the material state at the onset of damage, which is referred to as the damage initiation criterion. Beyond this point, the stress-strain response $c - d$ is governed by the evolution of the degradation of the stiffness in the region of strain localization. In the context of damage mechanics $c - d$ can be viewed as the degraded response of the curve $c - d'$ that the material would have followed in the absence of damage.

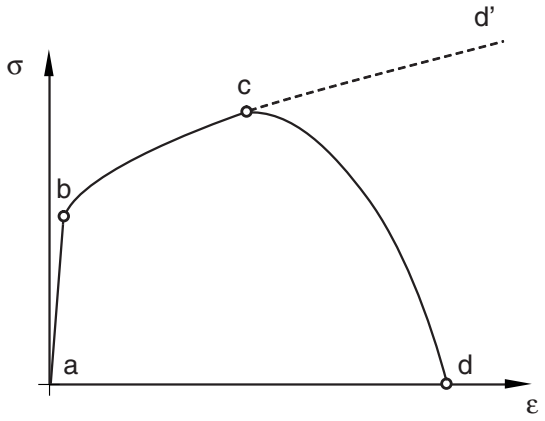


Figure 24.1.1–1 Typical uniaxial stress-strain response of a metal specimen.

Thus, in Abaqus the specification of a failure mechanism consists of four distinct parts:

- the definition of the effective (or undamaged) material response (e.g., $a - b - c - d'$ in Figure 24.1.1–1),
- a damage initiation criterion (e.g., ***c*** in Figure 24.1.1–1),
- a damage evolution law (e.g., $c - d$ in Figure 24.1.1–1), and
- a choice of element deletion whereby elements can be removed from the calculations once the material stiffness is fully degraded (e.g., ***d*** in Figure 24.1.1–1).

These parts will be discussed separately for ductile metals (“Damage and failure for ductile metals: overview,” Section 24.2.1) and fiber-reinforced materials (“Damage and failure for fiber-reinforced composites: overview,” Section 24.3.1).

Mesh dependency

In continuum mechanics the constitutive model is normally expressed in terms of stress-strain relations. When the material exhibits strain-softening behavior, leading to strain localization, this formulation results in a strong mesh dependency of the finite element results in that the energy dissipated decreases upon mesh refinement. In Abaqus all of the available damage evolution models use a formulation intended to alleviate the mesh dependency. This is accomplished by introducing a characteristic length into the formulation, which in Abaqus is related to the element size, and expressing the softening part of the constitutive law as a stress-displacement relation. In this case the energy dissipated during the damage process is specified per unit area, not per unit volume. This energy is treated as an additional material parameter, and it is used to compute the displacement at which full material damage occurs. This is consistent with the concept of critical energy release rate as a material parameter for fracture mechanics. This formulation ensures that the correct amount of energy is dissipated and greatly alleviates the mesh dependency.

24.2 Damage and failure for ductile metals

- “Damage and failure for ductile metals: overview,” Section 24.2.1
- “Damage initiation for ductile metals,” Section 24.2.2
- “Damage evolution and element removal for ductile metals,” Section 24.2.3

24.2.1 DAMAGE AND FAILURE FOR DUCTILE METALS: OVERVIEW

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Progressive damage and failure," Section 24.1.1
- "Damage initiation for ductile metals," Section 24.2.2
- "Damage evolution and element removal for ductile metals," Section 24.2.3
- *DAMAGE INITIATION
- *DAMAGE EVOLUTION
- "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Abaqus/Standard and Abaqus/Explicit offer a general capability for predicting the onset of failure and a capability for modeling progressive damage and failure of ductile metals. In the most general case this requires the specification of the following:

- the undamaged elastic-plastic response of the material ("Classical metal plasticity," Section 23.2.1);
- a damage initiation criterion ("Damage initiation for ductile metals," Section 24.2.2); and
- a damage evolution response, including a choice of element removal ("Damage evolution and element removal for ductile metals," Section 24.2.3).

A summary of the general framework for progressive damage and failure in Abaqus is given in "Progressive damage and failure," Section 24.1.1. This section provides an overview of the damage initiation criteria and damage evolution law for ductile metals. In addition, Abaqus/Explicit offers dynamic failure models that are suitable for high-strain-rate dynamic problems ("Dynamic failure models," Section 23.2.8).

Damage initiation criterion

Abaqus offers a variety of choices of damage initiation criteria for ductile metals, each associated with distinct types of material failure. They can be classified in the following categories:

- Damage initiation criteria for the fracture of metals, including ductile and shear criteria.
- Damage initiation criteria for the necking instability of sheet metal. These include forming limit diagrams (FLD, FLSD, and MSFLD) intended to assess the formability of sheet metal and the Marciniak-Kuczynski (M-K) criterion (available only in Abaqus/Explicit) to numerically predict necking instability in sheet metal taking into account the deformation history.

These criteria are discussed in "Damage initiation for ductile metals," Section 24.2.2. Each damage initiation criterion has an associated output variable to indicate whether the criterion has been met during the analysis. A value of 1.0 or higher indicates that the initiation criterion has been met.

More than one damage initiation criterion can be specified for a given material. If multiple damage initiation criteria are specified for the same material, they are treated independently. Once a particular initiation criterion is satisfied, the material stiffness is degraded according to the specified damage evolution law for that criterion; in the absence of a damage evolution law, however, the material stiffness is not degraded. A failure mechanism for which no damage evolution response is specified is said to be inactive. Abaqus will evaluate the initiation criterion for an inactive mechanism for output purposes only, but the mechanism will have no effect on the material response.

Input File Usage: Use the following option to define each damage initiation criterion (repeat as needed to define multiple criteria):

∗DAMAGE INITIATION, CRITERION=*criterion 1*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion***

Damage evolution

The damage evolution law describes the rate of degradation of the material stiffness once the corresponding initiation criterion has been reached. For damage in ductile metals Abaqus assumes that the degradation of the stiffness associated with each active failure mechanism can be modeled using a scalar damage variable, d_i ($i \in N_{\text{act}}$), where N_{act} represents the set of active mechanisms. At any given time during the analysis the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\bar{\boldsymbol{\sigma}},$$

where $\boldsymbol{D}$ is the overall damage variable and $\bar{\boldsymbol{\sigma}}$ is the effective (or undamaged) stress tensor computed in the current increment. $\bar{\boldsymbol{\sigma}}$ are the stresses that would exist in the material in the absence of damage. The material has lost its load-carrying capacity when $D = 1$. By default, an element is removed from the mesh if all of the section points at any one integration location have lost their load-carrying capacity.

The overall damage variable, $\boldsymbol{D}$, captures the combined effect of all active mechanisms and is computed in terms of the individual damage variables, d_i, according to a user-specified rule.

Abaqus supports different models of damage evolution in ductile metals and provides controls associated with element deletion due to material failure, as described in "Damage evolution and element removal for ductile metals," Section 24.2.3. All of the available models use a formulation intended to alleviate the strong mesh dependency of the results that can arise from strain localization effects during progressive damage.

Input File Usage: Use the following option immediately after the corresponding ∗DAMAGE INITIATION option to specify the damage evolution behavior:

∗DAMAGE EVOLUTION

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion*: Suboptions→Damage Evolution**

Elements

The failure modeling capability for ductile metals can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

For coupled temperature-displacement elements the thermal properties of the material are not affected by the progressive damage of the material stiffness until the condition for element deletion is reached; at this point the thermal contribution of the element is also removed.

The damage initiation criteria for sheet metal necking instability (FLD, FLSD, MSFLD, and M-K) are available only for elements that include mechanical behavior and use a plane stress formulation (i.e., plane stress, shell, continuum shell, and membrane elements).

24.2.2 DAMAGE INITIATION FOR DUCTILE METALS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Progressive damage and failure," Section 24.1.1
- *DAMAGE INITIATION
- "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The material damage initiation capability for ductile metals:

- is intended as a general capability for predicting initiation of damage in metals, including sheet, extrusion, and cast metals as well as other materials;
- can be used in combination with the damage evolution models for ductile metals described in "Damage evolution and element removal for ductile metals," Section 24.2.3;
- allows the specification of more than one damage initiation criterion;
- includes ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD) and Müschenborn-Sonne forming limit diagram (MSFLD) criteria for damage initiation;
- includes in Abaqus/Explicit the Marciniak-Kuczynski (M-K) and Johnson-Cook criteria for damage initiation;
- can be used in Abaqus/Standard in conjunction with Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity (ductile, shear, FLD, FLSD, and MSFLD criteria); and
- can be used in Abaqus/Explicit in conjunction with Mises and Johnson-Cook plasticity (ductile, shear, FLD, FLSD, MSFLD, Johnson-Cook, and MK criteria) and in conjunction with Hill and Drucker-Prager plasticity (ductile, shear, FLD, FLSD, MSFLD, and Johnson-Cook criteria).

Damage initiation criteria for fracture of metals

Two main mechanisms can cause the fracture of a ductile metal: ductile fracture due to the nucleation, growth, and coalescence of voids; and shear fracture due to shear band localization. Based on phenomenological observations, these two mechanisms call for different forms of the criteria for the onset of damage (Hooputra et al., 2004). The functional forms provided by Abaqus for these criteria are discussed below. These criteria can be used in combination with the damage evolution models for ductile metals discussed in "Damage evolution and element removal for ductile metals," Section 24.2.3, to model fracture of a ductile metal. (See "Progressive failure analysis of thin-wall aluminum extrusion under quasi-static and dynamic loads," Section 2.1.16 of the Abaqus Example Problems Manual, for an example.)

Ductile criterion

The ductile criterion is a phenomenological model for predicting the onset of damage due to nucleation, growth, and coalescence of voids. The model assumes that the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}^{pl}_{\mathrm{D}}$, is a function of stress triaxiality and strain rate:

$$\bar{\varepsilon}^{pl}_{\mathrm{D}}(\eta, \dot{\bar{\varepsilon}}^{pl}),$$

where $\eta = -p/q$ is the stress triaxiality, $\boldsymbol{p}$ is the pressure stress, $\boldsymbol{q}$ is the Mises equivalent stress, and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate. The criterion for damage initiation is met when the following condition is satisfied:

$$\omega_{\mathrm{D}} = \int \frac{\mathrm{d}\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_{\mathrm{D}}(\eta, \dot{\bar{\varepsilon}}^{pl})} = 1,$$

where ω_{D} is a state variable that increases monotonically with plastic deformation. At each increment during the analysis the incremental increase in ω_{D} is computed as

$$\Delta\omega_{\mathrm{D}} = \frac{\Delta\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_{\mathrm{D}}(\eta, \dot{\bar{\varepsilon}}^{pl})} \geq 0.$$

In Abaqus/Standard the ductile criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models and in Abaqus/Explicit in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state.

Input File Usage: Use the following option to specify the equivalent plastic strain at the onset of damage as a tabular function of stress triaxality, strain rate, and, optionally, temperature and predefined field variables:

∗DAMAGE INITIATION, CRITERION=DUCTILE, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→Ductile Damage**

Defining dependency of ductile criterion on Lode angle in Abaqus/Explicit

Recent experimental results for aluminum alloys and other metals (Bai and Wierzbicki, 2008) reveal that, in addition to stress triaxility and strain rate, ductile fracture can also depend on the third invariant of deviatoric stress, which is related to the Lode angle (or deviatoric polar angle). Abaqus/Explicit allows the definition of the equivalent plastic strain at the onset of ductile damage, $\bar{\varepsilon}^{pl}_{\mathrm{D}}$, as a function of the Lode angle, Θ, by way of the functional form

$$\bar{\varepsilon}^{pl}_{\mathrm{D}}(\eta, \xi(\Theta), \dot{\bar{\varepsilon}}^{pl}),$$

where

$$\xi(\Theta) = \cos(3\Theta) = \left(\frac{r}{q}\right)^3,$$

q is the Mises equivalent stress, and r is the third invariant of deviatoric stress, $r = \left(\frac{9}{2}\mathbf{S}\cdot\mathbf{S}:\mathbf{S}\right)^{\frac{1}{3}}$. The function $\xi(\Theta)$ can take values from $\xi = -1$, for stress states on the compressive meridian, to $\xi = 1$, for stress states on the tensile meridian.

Input File Usage: Use the following option to indicate that the equivalent plastic strain at the onset of ductile damage is a function of the Lode angle:

*DAMAGE INITIATION, CRITERION=DUCTILE, LODE DEPENDENT

Abaqus/CAE Usage: Defining dependency of ductile criterion on Lode angle is not supported in Abaqus/CAE.

Johnson-Cook criterion

The Johnson-Cook criterion (available only in Abaqus/Explicit) is a special case of the ductile criterion in which the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}^{pl}_{\mathrm{D}}$, is assumed to be of the form

$$\bar{\varepsilon}^{pl}_{\mathrm{D}} = \left[d_1 + d_2\exp\left(-d_3\eta\right)\right]\left[1 + d_4\ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right)\right]\left(1 + d_5\hat{\theta}\right),$$

where d_1–d_5 are failure parameters and $\dot{\varepsilon}_0$ is the reference strain rate. This expression differs from the original formula published by Johnson and Cook (1985) in the sign of the parameter d_3. This difference is motivated by the fact that most materials experience a decrease in $\bar{\varepsilon}^{pl}_{\mathrm{D}}$ with increasing stress triaxiality; therefore, d_3 in the above expression will usually take positive values. $\hat{\theta}$ is the nondimensional temperature defined as

$$\hat{\theta} \equiv \begin{cases} 0 & \text{for} \quad \theta < \theta_{\mathrm{transition}} \\ (\theta - \theta_{\mathrm{transition}})/(\theta_{\mathrm{melt}} - \theta_{\mathrm{transition}}) & \text{for} \quad \theta_{\mathrm{transition}} \le \theta \le \theta_{\mathrm{melt}} \\ 1 & \text{for} \quad \theta > \theta_{\mathrm{melt}} \end{cases},$$

where θ is the current temperature, θ_{melt} is the melting temperature, and $\theta_{\mathrm{transition}}$ is the transition temperature defined as the one at or below which there is no temperature dependence on the expression of the damage strain $\bar{\varepsilon}^{pl}_{\mathrm{D}}$. The material parameters must be measured at or below the transition temperature.

The Johnson-Cook criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state. When used in conjunction with the Johnson-Cook plasticity model, the specified values of the melting and transition temperatures should be consistent with the values specified in the plasticity definition. The Johnson-Cook damage initiation criterion can also be specified together with any other initiation criteria, including the ductile criteria; each initiation criterion is treated independently.

Input File Usage: Use the following option to specify the parameters for the Johnson-Cook initiation criterion:

*DAMAGE INITIATION, CRITERION=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→Johnson-Cook Damage**

Shear criterion

The shear criterion is a phenomenological model for predicting the onset of damage due to shear band localization. The model assumes that the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}_{\mathrm{S}}^{pl}$, is a function of the shear stress ratio and strain rate:

$$\bar{\varepsilon}_{\mathrm{S}}^{pl}(\theta_s, \dot{\bar{\varepsilon}}^{pl}).$$

Here $\theta_s = (q + k_s p)/\tau_{\max}$ is the shear stress ratio, $\tau_{\max}$ is the maximum shear stress, and k_s is a material parameter. A typical value of k_s for aluminum is k_s = 0.3 (Hooputra et al., 2004). The criterion for damage initiation is met when the following condition is satisfied:

$$\omega_S = \int \frac{\mathrm{d}\bar{\varepsilon}^{pl}}{\bar{\varepsilon}_S^{pl}(\theta_s, \dot{\bar{\varepsilon}}^{pl})} = 1,$$

where ω_{S} is a state variable that increases monotonically with plastic deformation proportional to the incremental change in equivalent plastic strain. At each increment during the analysis the incremental increase in ω_{S} is computed as

$$\Delta\omega_{\mathrm{S}} = \frac{\Delta\bar{\varepsilon}^{pl}}{\bar{\varepsilon}_{\mathrm{S}}^{pl}(\theta_s, \dot{\bar{\varepsilon}}^{pl})} \geq 0.$$

In Abaqus/Explicit the shear criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state. In Abaqus/Standard it can be used with the Mises, Johnson-Cook, Hill, and Drucker-Prager models.

Input File Usage: Use the following option to specify k_s and to specify the equivalent plastic strain at the onset of damage as a tabular function of the shear stress ratio, strain rate, and, optionally, temperature and predefined field variables:

*DAMAGE INITIATION, CRITERION=SHEAR, KS=k_s, DEPENDENCIES=n

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→Shear Damage**

Initial conditions

Optionally, you can specify the initial work hardened state of the material by providing the initial equivalent plastic strain values (see "Defining initial values of state variables for plastic hardening" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1) and, if residual stresses are also present, the initial stress values (see "Defining initial stresses" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). Abaqus uses this information to initialize the values of the ductile and shear damage initiation criteria, ω_{D} and ω_{S}, assuming constant values of stress triaxiality and shear shear ratio (linear stress path).

Input File Usage: Use the following options to specify that material hardening and residual stresses have occurred prior to the current analysis:

*INITIAL CONDITIONS, TYPE=HARDENING
*INITIAL CONDITIONS, TYPE=STRESS

Abaqus/CAE Usage: Use the following options to specify that material hardening and residual stresses have occurred prior to the current analysis:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** and **Stress** for the **Types for Selected Step**

Damage initiation criteria for sheet metal instability

Necking instability plays a determining factor in sheet metal forming processes: the size of the local neck region is typically of the order of the thickness of the sheet, and local necks can rapidly lead to fracture. Localized necking cannot be modeled with traditional shell elements used in sheet metal forming simulations because the size of the neck is of the order of the thickness of the element. Abaqus supports four criteria for predicting the onset of necking instability in sheet metals: forming limit diagram (FLD); forming limit stress diagram (FLSD); Müschenborn-Sonne forming limit diagram (MSFLD); and Marciniak-Kuczynski (M-K) criteria, which is available only in Abaqus/Explicit. These criteria apply only to elements with a plane stress formulation (plane stress, shell, continuum shell, and membrane elements); Abaqus ignores these criteria for other elements. The initiation criteria for necking instability can be used in combination with the damage evolution models discussed in "Damage evolution and element removal for ductile metals," Section 24.2.3, to account for the damage induced by necking.

Classical strain-based forming limit diagrams (FLDs) are known to be dependent on the strain path. Changes in the deformation mode (e.g., equibiaxial loading followed by uniaxial tensile strain) may result in major modifications in the level of the limit strains. Therefore, the FLD damage initiation criterion should be used with care if the strain paths in the analysis are nonlinear. In practical industrial applications, significant changes in the strain path may be induced by multistep forming operations, complex geometry of the tooling, and interface friction, among other factors. For problems with highly nonlinear strain paths Abaqus offers three additional damage initiation criteria: the forming limit stress diagram (FLSD) criterion, the Müschenborn-Sonne forming limit diagram (MSFLD) criterion, and in Abaqus/Explicit the Marciniak-Kuczynski (M-K) criterion; these alternatives to the FLD damage initiation criterion are intended to minimize load path dependence.

The characteristics of each criterion available in Abaqus for predicting damage initiation in sheet metals are discussed below.

Forming limit diagram (FLD) criterion

The forming limit diagram (FLD) is a useful concept introduced by Keeler and Backofen (1964) to determine the amount of deformation that a material can withstand prior to the onset of necking instability. The maximum strains that a sheet material can sustain prior to the onset of necking are referred to as the forming limit strains. A FLD is a plot of the forming limit strains in the space of principal (in-plane) logarithmic strains. In the discussion that follows *major* and *minor* limit strains refer to the

maximum and minimum values of the in-plane principal limit strains, respectively. The major limit strain is usually represented on the vertical axis and the minor strain on the horizontal axis, as illustrated in Figure 24.2.2–1. The line connecting the states at which deformation becomes unstable is referred to as the forming limit curve (FLC). The FLC gives a sense of the formability of a sheet of material. Strains computed numerically by Abaqus can be compared to a FLC to determine the feasibility of the forming process under analysis.

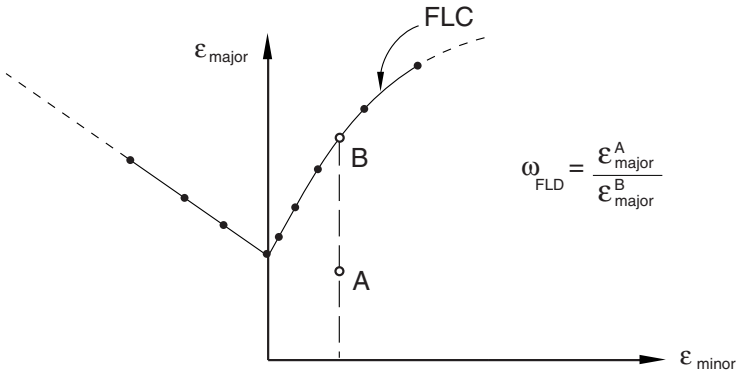


Figure 24.2.2–1 Forming limit diagram (FLD).

The FLD damage initiation criterion requires the specification of the FLC in tabular form by giving the major principal strain at damage initiation as a tabular function of the minor principal strain and, optionally, temperature and predefined field variables, $\varepsilon^{\mathrm{FLD}}_{\mathrm{major}}(\varepsilon_{\mathrm{minor}}, \theta, f_i)$. The damage initiation criterion for the FLD is given by the condition $\omega_{\mathrm{FLD}} = 1$, where the variable ω_{FLD} is a function of the current deformation state and is defined as the ratio of the current major principal strain, $\varepsilon_{\mathrm{major}}$, to the major limit strain on the FLC evaluated at the current values of the minor principal strain, $\varepsilon_{\mathrm{minor}}$; temperature, θ; and predefined field variables, f_i:

$$\omega_{\mathrm{FLD}} = \frac{\varepsilon_{\mathrm{major}}}{\varepsilon^{\mathrm{FLD}}_{\mathrm{major}}(\varepsilon_{\mathrm{minor}}, \theta, f_i)}.$$

For example, for the deformation state given by point A in Figure 24.2.2–1 the damage initiation criterion is evaluated as $\omega_{\mathrm{FLD}} = \varepsilon^{\mathrm{A}}_{\mathrm{major}}/\varepsilon^{\mathrm{B}}_{\mathrm{major}}$.

If the value of the minor strain lies outside the range of the specified tabular values, Abaqus will extrapolate the value of the major limit strain on the FLC by assuming that the slope at the endpoint of the curve remains constant. Extrapolation with respect to temperature and field variables follows the standard conventions: the property is assumed to be constant outside the specified range of temperature and field variables (see "Material data definition," Section 21.1.2).

Experimentally, FLDs are measured under conditions of biaxial stretching of a sheet, without bending effects. Under bending loading, however, most materials can achieve limit strains that are much greater than those on the FLC. To avoid the prediction of early failure under bending deformation, Abaqus evaluates the FLD criterion using the strains at the midplane through the thickness of the

element. For composite shells with several layers the criterion is evaluated at the midplane of each layer for which a FLD curve has been specified, which ensures that only biaxial stretching effects are taken into account. Therefore, the FLD criterion is not suitable for modeling failure under bending loading; other failure models (such as ductile and shear failure) are more appropriate for such loading. Once the FLD damage initiation criterion is met, the evolution of damage is driven independently at each material point through the thickness of the element based on the local deformation at that point. Thus, although bending effects do not affect the evaluation of the FLD criterion, they may affect the rate of evolution of damage.

Input File Usage: Use the following option to specify the limit major strain as a tabular function of minor strain:

*DAMAGE INITIATION, CRITERION=FLD

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→FLD Damage**

Forming limit stress diagram (FLSD) criterion

When strain-based FLCs are converted into stress-based FLCs, the resulting stress-based curves have been shown to be minimally affected by changes to the strain path (Stoughton, 2000); that is, different strain-based FLCs, corresponding to different strain paths, are mapped onto a single stress-based FLC. This property makes forming limit stress diagrams (FLSDs) an attractive alternative to FLDs for the prediction of necking instability under arbitrary loading. However, the apparent independence of the stress-based limit curves on the strain path may simply reflect the small sensitivity of the yield stress to changes in plastic deformation. This topic is still under discussion in the research community.

A FLSD is the stress counterpart of the FLD, with the major and minor principal in-plane stresses corresponding to the onset of necking localization plotted on the vertical and horizontal axes, respectively. In Abaqus the FLSD damage initiation criterion requires the specification of the major principal in-plane stress at damage initiation as a tabular function of the minor principal in-plane stress and, optionally, temperature and predefined field variables, $\sigma_{\mathrm{major}}^{\mathrm{FLSD}}(\sigma_{\mathrm{minor}}, \theta, f_i)$. The damage initiation criterion for the FLSD is met when the condition $\omega_{\mathrm{FLSD}} = 1$ is satisfied, where the variable ω_{FLSD} is a function of the current stress state and is defined as the ratio of the current major principal stress, σ_{major}, to the major stress on the FLSD evaluated at the current values of minor stress, σ_{minor}; temperature, θ; and predefined field variables, f_i:

$$\omega_{\mathrm{FLSD}} = \frac{\sigma_{\mathrm{major}}}{\sigma_{\mathrm{major}}^{\mathrm{FLSD}}(\sigma_{\mathrm{minor}}, \theta, f_i)}.$$

If the value of the minor stress lies outside the range of specified tabular values, Abaqus will extrapolate the value of the major limit stress assuming that the slope at the endpoints of the curve remains constant. Extrapolation with respect to temperature and field variables follows the standard conventions: the property is assumed to be constant outside the specified range of temperature and field variables (see "Material data definition," Section 21.1.2).

For reasons similar to those discussed earlier for the FLD criterion, Abaqus evaluates the FLSD criterion using the stresses averaged through the thickness of the element (or the layer, in the case of

composite shells with several layers), ignoring bending effects. Therefore, the FLSD criterion cannot be used to model failure under bending loading; other failure models (such as ductile and shear failure) are more suitable for such loading. Once the FLSD damage initiation criterion is met, the evolution of damage is driven independently at each material point through the thickness of the element based on the local deformation at that point. Thus, although bending effects do not affect the evaluation of the FLSD criterion, they may affect the rate of evolution of damage.

Input File Usage: Use the following option to specify the limit major stress as a tabular function of minor stress:

*DAMAGE INITIATION, CRITERION=FLSD

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→FLSD Damage**

Marciniak-Kuczynski (M-K) criterion

Another approach available in Abaqus/Explicit for accurately predicting the forming limits for arbitrary loading paths is based on the localization analysis proposed by Marciniak and Kuczynski (1967). The approach can be used with the Mises and Johnson-Cook plasticity models, including kinematic hardening. In M-K analysis, virtual thickness imperfections are introduced as grooves simulating preexisting defects in an otherwise uniform sheet material. The deformation field is computed inside each groove as a result of the applied loading outside the groove. Necking is considered to occur when the ratio of the deformation in the groove relative to the nominal deformation (outside the groove) is greater than a critical value.

Figure 24.2.2–2 shows schematically the geometry of the groove considered for M-K analysis. In the figure a denotes the nominal region in the shell element outside the imperfection, and b denotes the weak groove region. The initial thickness of the imperfection relative to the nominal thickness is given by the ratio $f_0 = t_0^b/t_0^a$, with the subscript 0 denoting quantities in the initial, strain-free state. The groove is oriented at a zero angle with respect to the 1-direction of the local material orientation.

Abaqus/Explicit allows the specification of an anisotropic distribution of thickness imperfections as a function of angle with respect to the local material orientation, $f_0(\theta)$. Abaqus/Explicit first solves for the stress-strain field in the nominal area ignoring the presence of imperfections; then it considers the effect of each groove alone. The deformation field inside each groove is computed by enforcing the strain compatibility condition

$$\varepsilon_{tt}^b = \varepsilon_{tt}^a$$

and the force equilibrium equations

$$F_{nn}^b = F_{nn}^a \quad \text{and}$$
$$F_{nt}^b = F_{nt}^a.$$

The subscripts n and t refer to the directions normal and tangential to the groove. In the above equilibrium equations F_{nn} and F_{nt} are forces per unit width in the t-direction.

The onset of necking instability is assumed to occur when the ratio of the rate of deformation inside a groove relative to the rate of deformation if no groove were present is greater than a critical value. In

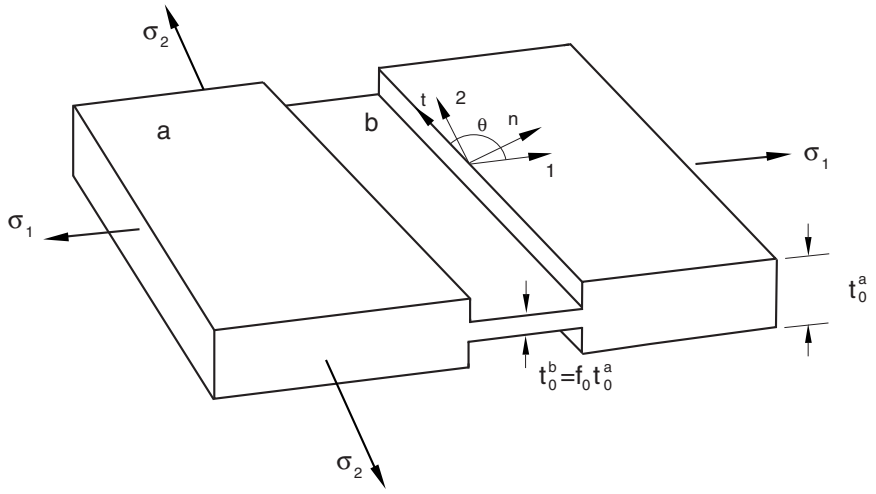


Figure 24.2.2–2 Imperfection model for the M-K analysis.

addition, it may not be possible to find a solution that satisfies equilibrium and compatibility conditions once localization initiates at a particular groove; consequently, failure to find a converged solution is also an indicator of the onset of localized necking. For the evaluation of the damage initiation criterion Abaqus/Explicit uses the following measures of deformation severity:

$$f_{eq} = \frac{\Delta\bar{\varepsilon}_b^{pl}}{\Delta\bar{\varepsilon}_a^{pl}},$$
$$f_{nn} = \frac{\Delta\varepsilon_{nn}^{b}}{\Delta\varepsilon_{nn}^{a}},$$
$$f_{nt} = \frac{\Delta\varepsilon_{nt}^{b}}{\Delta\varepsilon_{nt}^{a}}.$$

These deformation severity factors are evaluated on each of the specified groove directions and compared with the critical values. (The evaluation is performed only if the incremental deformation is primarily plastic; the M-K criterion will not predict damage initiation if the deformation increment is elastic.) The most unfavorable groove direction is used for the evaluation of the damage initiation criterion, which is given as

$$\omega_{\mathrm{MK}} = \max\left(\frac{f_{eq}}{f_{eq}^{\mathrm{crit}}}, \frac{f_{nn}}{f_{nn}^{\mathrm{crit}}}, \frac{f_{nt}}{f_{nt}^{\mathrm{crit}}}\right),$$

where f_{eq}^{crit}, f_{nn}^{crit}, and f_{nt}^{crit} are the critical values of the deformation severity indices. Damage initiation occurs when $\omega_{\mathrm{MK}} = 1$ or when a converged solution to the equilibrium and compatibility equations cannot be found. By default, Abaqus/Explicit assumes $f_{eq}^{\mathrm{crit}} = f_{nn}^{\mathrm{crit}} = f_{nt}^{\mathrm{crit}} = 10$; you can specify different values. If one of these parameters is set equal to zero, its corresponding deformation severity factor is not included in the evaluation of the damage initiation criterion. If all of these parameters are set

equal to zero, the M-K criterion is based solely on nonconvergence of the equilibrium and compatibility equations.

You must specify the fraction, f_0, equal to the initial thickness at the virtual imperfection divided by the nominal thickness (see Figure 24.2.2–2), as well as the number of imperfections to be used for the evaluation of the M-K damage initiation criterion. It is assumed that these directions are equally spaced angularly. By default, Abaqus/Explicit uses four imperfections located at 0°, 45°, 90°, and 135° with respect to the local 1-direction of the material. The initial imperfection size can be defined as a tabular function of angular direction, $f_0(\theta)$; this allows the modeling of an anisotropic distribution of flaws in the material. Abaqus/Explicit will use this table to evaluate the thickness of each of the imperfections that will be used for the evaluation of the M-K analysis method. In addition, the initial imperfection size can also be a function of initial temperature and field variables; this allows defining a nonuniform spatial distribution of imperfections. Abaqus/Explicit will compute the initial imperfection size based on the values of temperature and field variables at the beginning of the analysis. The initial size of the imperfection remains a constant property during the rest of the analysis.

A general recommendation is to choose the value of f_0 such that the forming limit predicted numerically for uniaxial strain loading conditions ($\varepsilon_{\text{minor}} = 0$) matches the experimental result.

The virtual grooves are introduced to evaluate the onset of necking instability; they do not influence the results in the underlying element. Once the criterion for necking instability is met, the material properties in the element are degraded according to the specified damage evolution law.

Input File Usage: Use the following option to specify the initial imperfection thickness relative to the nominal thickness as a tabular function of the angle with respect to the 1-direction of the local material orientation and, optionally, initial temperature and field variables:

∗DAMAGE INITIATION, CRITERION=MK, DEPENDENCIES=*n*

Use the following option to specify critical deformation severity factors:

∗DAMAGE INITIATION, CRITERION=MK, FEQ=f_{eq}^{crit}, FNN=f_{nn}^{crit}, FNT=f_{nt}^{crit}

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→M-K Damage**

Performance considerations for the M-K criterion

There can be a substantial increase in the overall computational cost when the M-K criterion is used. For example, the cost of processing a shell element with three section points through the thickness and four imperfections, which is the default for the M-K criterion, increases by approximately a factor of two compared to the cost without the M-K criterion. You can mitigate the cost of evaluating this damage initiation criterion by reducing the number of flaw directions considered or by increasing the number of increments between M-K computations, as explained below. Of course, the effect on the overall analysis cost depends on the fraction of the elements in the model that use this damage initiation criterion. The computational cost per element with the M-K criterion increases by approximately a factor of

$$1 + 0.25\frac{n_{\text{imp}}}{N_{\text{incr}}},$$

where n_{imp} is the number of imperfections specified for the evaluation of the M-K criterion and N_{incr} is the frequency, in number of increments, at which the M-K computations are performed. The coefficient of 0.25 in the above formula gives a reasonable estimate of the cost increase in most cases, but the actual cost increase may vary from this estimate. By default, Abaqus/Explicit performs the M-K computations on each imperfection at each time increment, $N_{\text{incr}} = 1$. Care must be taken to ensure that the M-K computations are performed frequently enough to ensure the accurate integration of the deformation field on each imperfection.

Input File Usage: Use the following option to specify the number of imperfections and frequency of the M-K analysis:

∗DAMAGE INITIATION, CRITERION=MK,
NUMBER IMPERFECTIONS=n_{imp}, FREQUENCY= N_{incr}

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→M-K Damage**: **Number of imperfections** and **Frequency**

Müschenborn-Sonne forming limit diagram (MSFLD) criterion

Müschenborn and Sonne (1975) proposed a method to predict the influence of the deformation path on the forming limits of sheet metals on the basis of the equivalent plastic strain, by assuming that the forming limit curve represents the sum of the highest attainable equivalent plastic strains. Abaqus makes use of a generalization of this idea to establish a criterion of necking instability of sheet metals for arbitrary deformation paths. The approach requires transforming the original forming limit curve (without predeformation effects) from the space of major versus minor strains to the space of equivalent plastic strain, $\bar{\varepsilon}^{pl}$, versus ratio of principal strain rates, $\alpha = \dot{\varepsilon}_{\text{minor}}/\dot{\varepsilon}_{\text{major}}$.

For linear strain paths, assuming plastic incompressibility and neglecting elastic strains:

$$\alpha = \varepsilon_{\text{minor}}/\varepsilon_{\text{major}} \quad \text{and}$$
$$\bar{\varepsilon}^{pl} = \varepsilon_{\text{major}}\sqrt{\frac{4}{3}(1 + \alpha + \alpha^2)}.$$

As illustrated in Figure 24.2.2–3 , linear deformation paths in the FLD transform onto vertical paths in the $\bar{\varepsilon}^{pl}$–α diagram (constant value of α).

According to the MSFLD criterion, the onset of localized necking occurs when the sequence of deformation states in the $\bar{\varepsilon}^{pl}$–α diagram intersects the forming limit curve, as discussed below. It is emphasized that for linear deformation paths both FLD and MSFLD representations are identical and give rise to the same predictions. For arbitrary loading, however, the MSFLD representation takes into account the effects of the history of deformation through the use of the accumulated equivalent plastic strain.

For the specification of the MSFLD damage initiation criterion in Abaqus, you can directly provide the equivalent plastic strain at damage initiation as a tabular function of α and, optionally, equivalent plastic strain rate, temperature, and predefined field variables, $\bar{\varepsilon}^{pl}_{\text{MSFLD}}(\alpha, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i)$. Alternatively, you

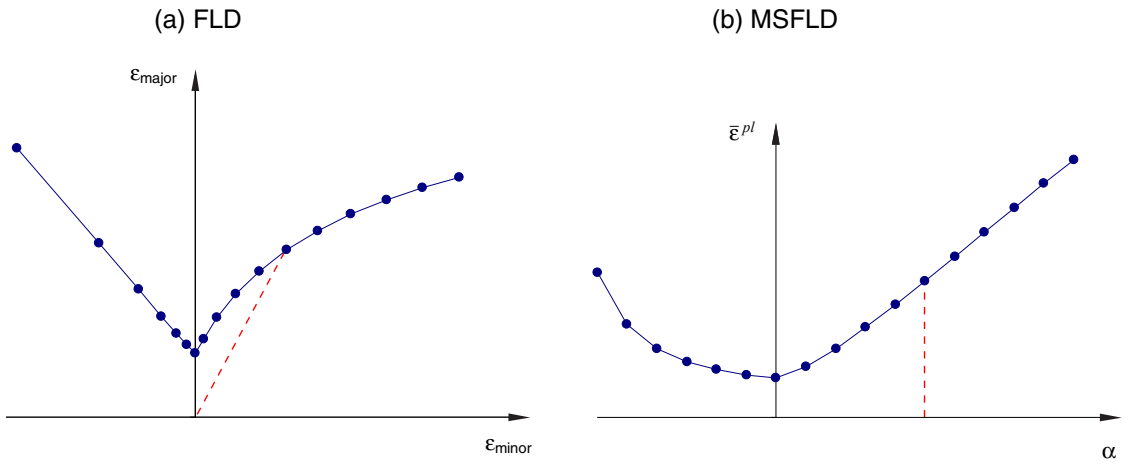


Figure 24.2.2–3 Transformation of the forming limit curve from traditional FLD representation (a) to MSFLD representation (b). Linear deformation paths transform onto vertical paths.

can specify the curve in the traditional FLD format (in the space of major and minor strains) by providing a tabular function of the form $\varepsilon_{\text{major}}(\varepsilon_{\text{minor}}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i)$. In this case Abaqus will automatically transform the data into the $\bar{\varepsilon}^{pl}$–α format.

Let ω_{MSFLD} represent the ratio of the current equivalent plastic strain, $\bar{\varepsilon}^{pl}$, to the equivalent plastic strain on the limit curve evaluated at the current values of α; strain rate, $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and predefined field variables, f_i:

$$\omega_{\text{MSFLD}} = \frac{\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_{\text{MSFLD}}(\alpha, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i)}.$$

The MSFLD criterion for necking instability is met when the condition $\omega_{\text{MSFLD}} = 1$ is satisfied. Necking instability also occurs if the sequence of deformation states in the $\bar{\varepsilon}^{pl}$–α diagram intersects the limit curve due to a sudden change in the straining direction. This situation is illustrated in Figure 24.2.2–4. As α changes from α_t to $\alpha_{t+\Delta t}$, the line connecting the corresponding points in the $\bar{\varepsilon}^{pl}$–α diagram intersects with the forming limit curve. When this situation occurs, the MSFLD criterion is reached despite the fact that $\bar{\varepsilon}^{pl}_{t+\Delta t} < \bar{\varepsilon}^{pl}_{\text{MSFLD}}(\alpha_{t+\Delta t})$. For output purposes Abaqus sets the value of ω_{MSFLD} equal to one to indicate that the criterion has been met.

The equivalent plastic strain $\bar{\varepsilon}^{pl}$ used for the evaluation of the MSFLD criterion in Abaqus is accumulated only over increments that result in an increase of the element area. Strain increments associated with a reduction of the element area cannot cause necking and do not contribute toward the evaluation of the MSFLD criterion.

If the value of α lies outside the range of specified tabular values, Abaqus extrapolates the value of equivalent plastic strain for initiation of necking assuming that the slope at the endpoints of the curve remains constant. Extrapolation with respect to strain rate, temperature, and field variables follows the

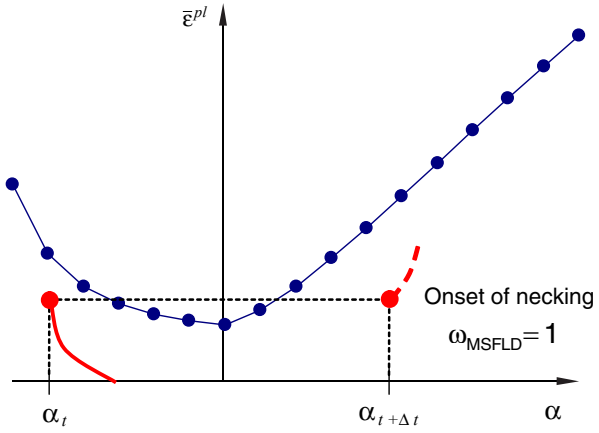


Figure 24.2.2–4 Illustration of how a sudden change in the straining direction, from α_t to $\alpha_{t+\Delta t}$, can produce a horizontal intersection with the limit curve and lead to onset of necking.

standard conventions: the property is assumed to be constant outside the specified range of strain rate, temperature, and field variables (see "Material data definition," Section 21.1.2).

As discussed in "Progressive damage and failure of ductile metals," Section 2.2.21 of the Abaqus Verification Manual, predictions of necking instability based on the MSFLD criterion agree remarkably well with predictions based on the Marciniak and Kuczynski criterion, at significantly less computational cost than the Marciniak and Kuczynski criterion. There are some situations, however, in which the MSFLD criterion may overpredict the amount of formability left in the material. This occurs in situations when, sometime during the loading history, the material reaches a state that is very close to the point of necking instability and is subsequently strained in a direction along which it can sustain further deformation. In this case the MSFLD criterion may predict that the amount of additional formability in the new direction is greater than that predicted with the Marciniak and Kuczynski criterion. However, this situation is often not a concern in practical forming applications where safety factors in the forming limit diagrams are commonly used to ensure that the material state is sufficiently far away from the point of necking. Refer to "Progressive damage and failure of ductile metals," Section 2.2.21 of the Abaqus Verification Manual, for a comparative analysis of these two criteria.

For reasons similar to those discussed earlier for the FLD criterion, Abaqus evaluates the MSFLD criterion using the strains at the midplane through the thickness of the element (or the layer, in the case of composite shells with several layers), ignoring bending effects. Therefore, the MSFLD criterion cannot be used to model failure under bending loading; other failure models (such as ductile and shear failure) are more suitable for such loading. Once the MSFLD damage initiation criterion is met, the evolution of damage is driven independently at each material point through the thickness of the element based on the local deformation at that point. Thus, although bending effects do not affect the evaluation of the MSFLD criterion, they may affect the rate of evolution of damage.

Input File Usage: Use the following option to specify the MSFLD damage initiation criterion by providing the limit equivalent plastic strain as a tabular function of α (default):

*DAMAGE INITIATION, CRITERION=MSFLD, DEFINITION=MSFLD

Use the following option to specify the MSFLD damage initiation criterion by providing the limit major strain as a tabular function of minor strain:

*DAMAGE INITIATION, CRITERION=MSFLD, DEFINITION=FLD

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→MSFLD Damage**

Numerical evaluation of the principal strain rates ratio

The ratio of principal strain rates, $\alpha = \dot{\varepsilon}_{\text{minor}}/\dot{\varepsilon}_{\text{major}}$, can jump in value due to sudden changes in the deformation path. Special care is required during explicit dynamic simulations to avoid nonphysical jumps in α triggered by numerical noise, which may cause a horizontal intersection of the deformation state with the forming limit curve and lead to the premature prediction of necking instability.

To overcome this problem, rather than computing α as a ratio of instantaneous strain rates, Abaqus/Explicit periodically updates α based on accumulated strain increments after small but significant changes in the equivalent plastic strain. The threshold value for the change in equivalent plastic strain triggering an update of α is denoted as $(\Delta\bar{\varepsilon}^{pl})^*$, and α is approximated as

$$\alpha = \frac{(\Delta\varepsilon)^*_{\text{minor}}}{(\Delta\varepsilon)^*_{\text{major}}},$$

where $(\Delta\varepsilon)^*_{\text{minor}}$ and $(\Delta\varepsilon)^*_{\text{major}}$ are principal values of the accumulated plastic strain since the previous update of α. The default value of $(\Delta\bar{\varepsilon}^{pl})^*$ is 0.002 (0.2%).

In addition, Abaqus/Explicit supports the following filtering method for the computation of α:

$$\alpha_{t+(\Delta t)^*} = (1-\omega)\alpha_t + \omega\frac{(\Delta\varepsilon)^*_{\text{minor}}}{(\Delta\varepsilon)^*_{\text{major}}},$$

where $(\Delta t)^*$ represents the accumulated time over the analysis increments required to have an increase in equivalent plastic strain of at least $(\Delta\bar{\varepsilon}^{pl})^*$. The factor ω $(0 < \omega \leq 1)$ facilitates filtering high-frequency oscillations. This filtering method is usually not necessary provided that an appropriate value of $(\Delta\bar{\varepsilon}^{pl})^*$ is used. You can specify the value of ω directly. The default value is $\omega = 1.0$ (no filtering).

In Abaqus/Standard α is computed at every analysis increment as $\alpha_{t+\Delta t} = \Delta\varepsilon_{\text{minor}}/\Delta\varepsilon_{\text{major}}$, without using either of the above filtering methods. However, you can still specify values for $(\Delta\bar{\varepsilon}^{pl})^*$ and ω; and these values can be imported into any subsequent analysis in Abaqus/Explicit.

Input File Usage: *DAMAGE INITIATION, CRITERION=MSFLD, PEINC=$(\Delta\bar{\varepsilon}^{pl})^*$, OMEGA=$\omega$

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→MSFLD Damage**: **Omega:** ω

The value for $(\Delta\bar{\varepsilon}^{pl})^*$ cannot be specified directly in Abaqus/CAE.

Initial conditions

When we need to study the behavior of a material that has been previously subjected to deformations, such as those originated during the manufacturing process, initial equivalent plastic strain values can be provided to specify the initial work hardened state of the material (see "Defining initial values of state variables for plastic hardening" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

In addition, when the initial equivalent plastic strain is greater than the minimum value on the forming limit curve, the initial value of α plays an important role in determining whether the MSFLD damage initiation criterion will be met during subsequent deformation. It is, therefore, important to specify the initial value of α in these situations. To this end, you can specify initial values of the plastic strain tensor (see "Defining initial values of plastic strain" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). Abaqus will use this information to compute the initial value of α as the ratio of the minor and major principal plastic strains; that is, neglecting the elastic component of deformation and assuming a linear deformation path.

Input File Usage: Use both of the following options to specify that material hardening and plastic strain have occurred prior to the current analysis:

*INITIAL CONDITIONS, TYPE=HARDENING
*INITIAL CONDITIONS, TYPE=PLASTIC STRAIN

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Initial plastic strain conditions are not supported in Abaqus/CAE.

Elements

The damage initiation criteria for ductile metals can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom) except for the pipe elements in Abaqus/Explicit.

The models for sheet metal necking instability (FLD, FLSD, MSFLD, and M-K) are available only with elements that include mechanical behavior and use a plane stress formulation (i.e., plane stress, shell, continuum shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus ("Output variables," Section 4.2), the following variables have special meaning when a damage initiation criterion is specified:

ERPRATIO	Ratio of principal strain rates, α, used for the MSFLD damage initiation criterion.
SHRRATIO	Shear stress ratio, $\theta_s = (q+k_s p)/\tau_{\max}$, used for the evaluation of the shear damage initiation criterion.
TRIAX	Stress triaxiality, $\eta = -p/q$ (available in Abaqus/Standard only in conjunction with damage initiation).

DMICRT	All damage initiation criteria components listed below.
DUCTCRT	Ductile damage initiation criterion, ω_{D}.
JCCRT	Johnson-Cook damage initiation criterion (available only in Abaqus/Explicit).
SHRCRT	Shear damage initiation criterion, ω_{S}.
FLDCRT	Maximum value of the FLD damage initiation criterion, ω_{FLD}, during the analysis.
FLSDCRT	Maximum value of the FLSD damage initiation criterion, ω_{FSLD}, during the analysis.
MSFLDCRT	Maximum value of the MSFLD damage initiation criterion, ω_{MSFLD}, during the analysis.
MKCRT	Marciniak-Kuczynski damage initiation criterion (available only in Abaqus/Explicit), ω_{MK}.

A value of 1 or greater for output variables associated with a damage initiation criterion indicates that the criterion has been met. Abaqus will limit the maximum value of the output variable to 1 if a damage evolution law has been prescribed for that criterion (see "Damage evolution and element removal for ductile metals," Section 24.2.3). However, if no damage evolution is specified, the criterion for damage initiation will continue to be computed beyond the point of damage initiation; in this case the output variable can take values greater than 1, indicating by how much the initiation criterion has been exceeded.

Additional references

- Hooputra, H., H. Gese, H. Dell, and H. Werner, "A Comprehensive Failure Model for Crashworthiness Simulation of Aluminium Extrusions," International Journal of Crashworthiness, vol. 9, no. 5, pp. 449–464, 2004.
- Bai, Y., and T. Wierzbicki, "A New Model of Metal Plasticity and Fracture with Pressure and Lode Dependence," International Journal of Plasticity, vol. 24, no. 6, pp. 1071–1096, 2008.
- Johnson, G. R., and W. H. Cook, "Fracture Characteristics of Three Metals Subjected to Various Strains, Strain rates, Temperatures and Pressures," Engineering Fracture Mechanics, vol. 21, no. 1, pp. 31–48, 1985.
- Keeler, S. P., and W. A. Backofen, "Plastic Instability and Fracture in Sheets Stretched over Rigid Punches," ASM Transactions Quarterly, vol. 56, pp. 25–48, 1964.
- Marciniak, Z., and K. Kuczynski, "Limit Strains in the Processes of Stretch Forming Sheet Metal," International Journal of Mechanical Sciences, vol. 9, pp. 609–620, 1967.
- Müschenborn, W., and H. Sonne, "Influence of the Strain Path on the Forming Limits of Sheet Metal," Archiv fur das Eisenhüttenwesen, vol. 46, no. 9, pp. 597–602, 1975.
- Stoughton, T. B., "A General Forming Limit Criterion for Sheet Metal Forming," International Journal of Mechanical Sciences, vol. 42, pp. 1–27, 2000.

24.2.3 DAMAGE EVOLUTION AND ELEMENT REMOVAL FOR DUCTILE METALS

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Progressive damage and failure," Section 24.1.1
- *DAMAGE EVOLUTION
- "Damage evolution" in "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The damage evolution capability for ductile metals:

- assumes that damage is characterized by the progressive degradation of the material stiffness, leading to material failure;
- must be used in combination with a damage initiation criterion for ductile metals ("Damage initiation for ductile metals," Section 24.2.2);
- uses mesh-independent measures (either plastic displacement or physical energy dissipation) to drive the evolution of damage after damage initiation;
- takes into account the combined effect of different damage mechanisms acting simultaneously on the same material and includes options to specify how each mechanism contributes to the overall material degradation; and
- offers options for what occurs upon failure, including the removal of elements from the mesh.

Damage evolution

Figure 24.2.3–1 illustrates the characteristic stress-strain behavior of a material undergoing damage. In the context of an elastic-plastic material with isotropic hardening, the damage manifests itself in two forms: softening of the yield stress and degradation of the elasticity. The solid curve in the figure represents the damaged stress-strain response, while the dashed curve is the response in the absence of damage. As discussed later, the damaged response depends on the element dimensions such that mesh dependency of the results is minimized.

In the figure σ_{y0} and $\bar{\varepsilon}_0^{pl}$ are the yield stress and equivalent plastic strain at the onset of damage, and $\bar{\varepsilon}_f^{pl}$ is the equivalent plastic strain at failure; that is, when the overall damage variable reaches the value $D = 1$. The overall damage variable, $\boldsymbol{D}$, captures the combined effect of all active damage mechanisms and is computed in terms of the individual damage variables, d_i, as discussed later in this section (see "Evaluating overall damage when multiple criteria are active").

The value of the equivalent plastic strain at failure, $\bar{\varepsilon}_f^{pl}$, depends on the characteristic length of the element and cannot be used as a material parameter for the specification of the damage evolution law.

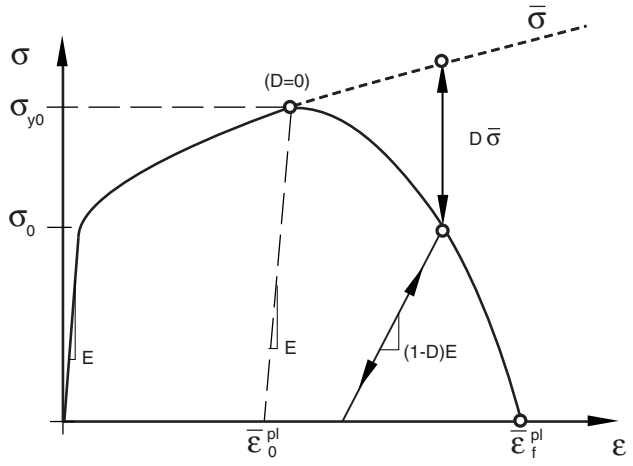


Figure 24.2.3–1 Stress-strain curve with progressive damage degradation.

Instead, the damage evolution law is specified in terms of equivalent plastic displacement, $\bar{u}^{pl}$, or in terms of fracture energy dissipation, G_f; these concepts are defined next.

Mesh dependency and characteristic length

When material damage occurs, the stress-strain relationship no longer accurately represents the material's behavior. Continuing to use the stress-strain relation introduces a strong mesh dependency based on strain localization, such that the energy dissipated decreases as the mesh is refined. A different approach is required to follow the strain-softening branch of the stress-strain response curve. Hillerborg's (1976) fracture energy proposal is used to reduce mesh dependency by creating a stress-displacement response after damage is initiated. Using brittle fracture concepts, Hillerborg defines the energy required to open a unit area of crack, G_f, as a material parameter. With this approach, the softening response after damage initiation is characterized by a stress-displacement response rather than a stress-strain response.

The implementation of this stress-displacement concept in a finite element model requires the definition of a characteristic length, $\boldsymbol{L}$, associated with an integration point. The fracture energy is then given as

$$G_f = \int_{\bar{\varepsilon}_0^{pl}}^{\bar{\varepsilon}_f^{pl}} L\sigma_y d\bar{\varepsilon}^{pl} = \int_0^{\bar{u}_f^{pl}} \sigma_y d\bar{u}^{pl}.$$

This expression introduces the definition of the equivalent plastic displacement, $\bar{u}^{pl}$, as the fracture work conjugate of the yield stress after the onset of damage (work per unit area of the crack). Before damage initiation $\dot{\bar{u}}^{pl} = 0$; after damage initiation $\dot{\bar{u}}^{pl} = L\dot{\bar{\varepsilon}}^{pl}$.

The definition of the characteristic length depends on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for

a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic length is used because the direction in which fracture occurs is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect, and elements that have aspect ratios close to unity are recommended.

Each damage initiation criterion described in "Damage initiation for ductile metals," Section 24.2.2, may have an associated damage evolution law. The damage evolution law can be specified in terms of equivalent plastic displacement, $\bar{u}^{pl}$, or in terms of fracture energy dissipation, G_f. Both of these options take into account the characteristic length of the element to alleviate mesh dependency of the results.

Evaluating overall damage when multiple criteria are active

The overall damage variable, D, captures the combined effect of all active mechanisms and is computed in terms of individual damage variables, d_i, for each mechanism. You can choose to combine some of the damage variables in a multiplicative sense to form an intermediate variable, d_{mult}, as follows:

$$d_{\text{mult}} = 1 - \prod_{k \in N_{\text{mult}}} (1 - d_k).$$

Then, the overall damage variable is computed as the maximum of d_{mult} and the remaining damage variables:

$$D = \max \left\{ d_{\text{mult}}, \max_{j \in N_{\text{max}}} (d_j) \right\}.$$

In the above expressions N_{mult} and N_{max} represent the sets of active mechanisms that contribute to the overall damage in a multiplicative and a maximum sense, respectively, with $N_{\text{act}} = N_{\text{mult}} \cup N_{\text{max}}$.

Input File Usage: Use the following option to specify that the damage associated with a particular criterion contributes to the overall damage variable in a maximum sense (default):

*DAMAGE EVOLUTION, DEGRADATION=MAXIMUM

Use the following option to specify that the damage associated with a particular criterion contributes to the overall damage variable in a multiplicative sense:

*DAMAGE EVOLUTION, DEGRADATION=MULTIPLICATIVE

Abaqus/CAE Usage: Use the following options to specify that the damage associated with a particular criterion contributes to the overall damage variable in a maximum sense (default) or in a multiplicative sense, respectively:

Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion***: **Suboptions→Damage Evolution**: **Degradation: Maximum** or **Multiplicative**

Defining damage evolution based on effective plastic displacement

As discussed previously, once the damage initiation criterion has been reached, the effective plastic displacement, $\bar{u}^{pl}$, is defined with the evolution equation

$$\dot{\bar{u}}^{pl} = L\dot{\bar{\varepsilon}}^{pl},$$

where L is the characteristic length of the element.

The evolution of the damage variable with the relative plastic displacement can be specified in tabular, linear, or exponential form. Instantaneous failure will occur if the plastic displacement at failure, $\bar{u}_f^{pl}$, is specified as 0; however, this choice is not recommended and should be used with care because it causes a sudden drop of the stress at the material point that can lead to dynamic instabilities.

Tabular form

You can specify the damage variable directly as a tabular function of equivalent plastic displacement, $d = d(\bar{u}^{pl})$, as shown in Figure 24.2.3–2(a).

Input File Usage: *DAMAGE EVOLUTION, TYPE=DISPLACEMENT, SOFTENING=TABULAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion*: Suboptions→Damage Evolution**: **Type: Displacement**: **Softening: Tabular**

Linear form

Assume a linear evolution of the damage variable with effective plastic displacement, as shown in Figure 24.2.3–2(b). You can specify the effective plastic displacement, $\bar{u}_f^{pl}$, at the point of failure (full degradation). Then, the damage variable increases according to

$$\dot{d} = \frac{L\dot{\bar{\varepsilon}}^{pl}}{\bar{u}_f^{pl}} = \frac{\dot{\bar{u}}^{pl}}{\bar{u}_f^{pl}}.$$

This definition ensures that when the effective plastic displacement reaches the value $\bar{u}^{pl} = \bar{u}_f^{pl}$, the material stiffness will be fully degraded ($d = 1$). The linear damage evolution law defines a truly linear stress-strain softening response only if the effective response of the material is perfectly plastic (constant yield stress) after damage initiation.

Input File Usage: *DAMAGE EVOLUTION, TYPE=DISPLACEMENT, SOFTENING=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion*: Suboptions→Damage Evolution**: **Type: Displacement**: **Softening: Linear**

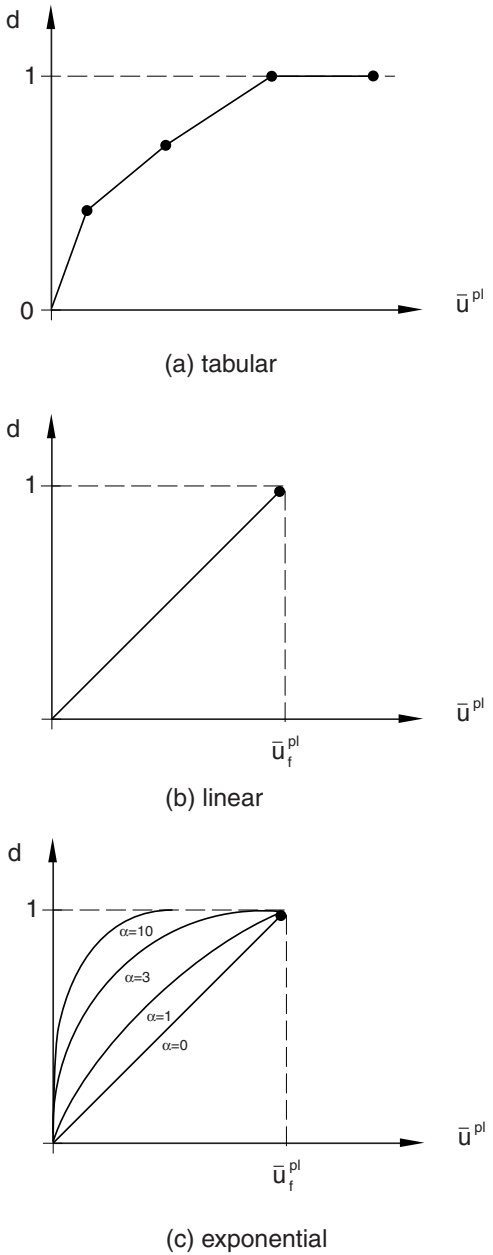


Figure 24.2.3–2 Different definitions of damage evolution based on plastic displacement: (a) tabular, (b) linear, and (c) exponential.

Exponential form

Assume an exponential evolution of the damage variable with plastic displacement, as shown in Figure 24.2.3–2(c). You can specify the relative plastic displacement at failure, $\bar{u}_f^{pl}$, and the exponent α. The damage variable is given as

$$d = \frac{1 - e^{-\alpha(\bar{u}^{pl}/\bar{u}_f^{pl})}}{1 - e^{-\alpha}}.$$

Input File Usage: *DAMAGE EVOLUTION, TYPE=DISPLACEMENT, SOFTENING=EXPONENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion*: Suboptions→Damage Evolution: Type: Displacement: Softening: Exponential**

Defining damage evolution based on energy dissipated during the damage process

You can specify the fracture energy per unit area, G_f, to be dissipated during the damage process directly. Instantaneous failure will occur if G_f is specified as 0. However, this choice is not recommended and should be used with care because it causes a sudden drop in the stress at the material point that can lead to dynamic instabilities.

The evolution in the damage can be specified in linear or exponential form.

Linear form

Assume a linear evolution of the damage variable with plastic displacement. You can specify the fracture energy per unit area, G_f. Then, once the damage initiation criterion is met, the damage variable increases according to

$$\dot{d} = \frac{L\dot{\bar{\varepsilon}}^{pl}}{\bar{u}_f^{pl}} = \frac{\dot{\bar{u}}^{pl}}{\bar{u}_f^{pl}},$$

where the equivalent plastic displacement at failure is computed as

$$\bar{u}_f^{pl} = \frac{2G_f}{\sigma_{y0}}$$

and σ_{y0} is the value of the yield stress at the time when the failure criterion is reached. Therefore, the model becomes equivalent to that shown in Figure 24.2.3–2(b). The model ensures that the energy dissipated during the damage evolution process is equal to G_f only if the effective response of the material is perfectly plastic (constant yield stress) beyond the onset of damage.

Input File Usage: *DAMAGE EVOLUTION, TYPE=ENERGY, SOFTENING=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion***: **Suboptions→Damage Evolution**: **Type: Energy**: **Softening: Linear**

Exponential form

Assume an exponential evolution of the damage variable given as

$$d = 1 - \exp\left(-\int_0^{\bar{u}^{pl}} \frac{\bar{\sigma}_y \dot{\bar{u}}^{pl}}{G_f}\right).$$

The formulation of the model ensures that the energy dissipated during the damage evolution process is equal to G_f, as shown in Figure 24.2.3–3(a). In theory, the damage variable reaches a value of 1 only asymptotically at infinite equivalent plastic displacement (Figure 24.2.3–3(b)). In practice, Abaqus/Explicit will set d equal to one when the dissipated energy reaches a value of $0.99G_f$.

Input File Usage: *DAMAGE EVOLUTION, TYPE=ENERGY, SOFTENING=EXPONENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Ductile Metals→*criterion***: **Suboptions→Damage Evolution**: **Type: Energy**: **Softening: Exponential**

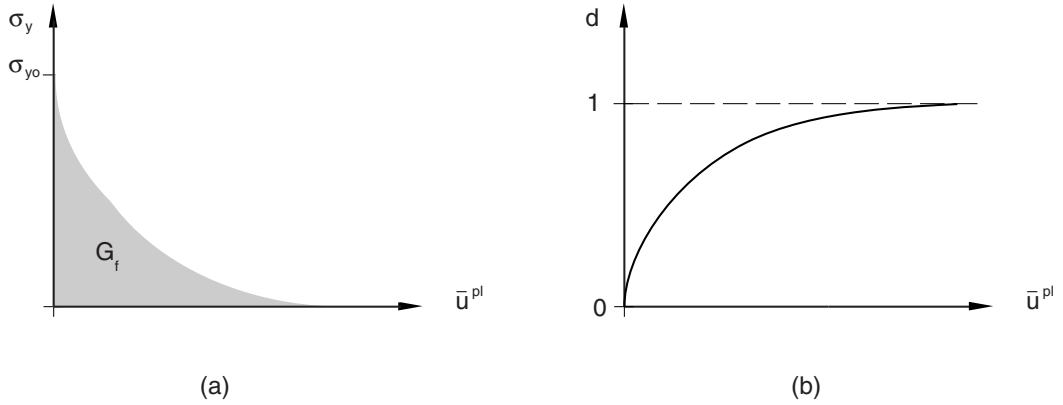


Figure 24.2.3–3 Energy-based damage evolution with exponential law: evolution of (a) yield stress and (b) damage variable.

Maximum degradation and choice of element removal

You have control over how Abaqus treats elements with severe damage. You can specify an upper bound, $D_{\max}$, to the overall damage variable, D; and you can choose whether to delete an element once maximum degradation is reached. The latter choice also affects which stiffness components are damaged.

Specifying the value of maximum degradation

The default setting of $D_{\max}$ depends on whether elements are to be deleted upon reaching maximum degradation (discussed next). For the default case of element deletion and in all cases for cohesive elements, $D_{\max} = 1.0$; otherwise, $D_{\max} = 0.99$. The output variable SDEG contains the value of $\boldsymbol{D}$. No further damage is accumulated at an integration point once $\boldsymbol{D}$ reaches $D_{\max}$ (except, of course, any remaining stiffness is lost upon element deletion).

Input File Usage: Use the following option to specify $D_{\max}$:

*SECTION CONTROLS, MAX DEGRADATION=$D_{\max}$

Removing the element from the mesh

Elements are deleted by default upon reaching maximum degradation. Except for cohesive elements with traction-separation response (see "Defining the constitutive response of cohesive elements using a traction-separation description," Section 32.5.6), Abaqus applies damage to all stiffness components equally for elements that may eventually be removed:

$$\boldsymbol{\sigma} = (1 - D)\bar{\boldsymbol{\sigma}}.$$

In Abaqus/Standard an element is removed from the mesh if $\boldsymbol{D}$ reaches $D_{\max}$ at all of the section points at all the integration locations of an element except for cohesive elements (for cohesive elements the conditions for element deletion are that $\boldsymbol{D}$ reaches $D_{\max}$ at all integration points and, for traction-separation response, none of the integration points are in compression).

In Abaqus/Explicit an element is removed from the mesh if $\boldsymbol{D}$ reaches $D_{\max}$ at all of the section points at any one integration location of an element except for cohesive elements (for cohesive elements the conditions for element deletion are that $\boldsymbol{D}$ reaches $D_{\max}$ at all integration points and, for traction-separation response, none of the integration points are in compression). For example, removal of a solid element takes place, by default, when maximum degradation is reached at any one integration point. However, in a shell element all through-the-thickness section points at any one integration location of an element must fail before the element is removed from the mesh. In the case of second-order reduced-integration beam elements, reaching maximum degradation at all section points through the thickness at either of the two element integration locations along the beam axis leads, by default, to element removal. Similarly, in modified triangular and tetrahedral solid elements and fully integrated membrane elements $\boldsymbol{D}$ reaching $D_{\max}$ at any one integration point leads, by default, to element removal.

In a heat transfer analysis the thermal properties of the material are not affected by the progressive damage of the material stiffness until the condition for element deletion is reached; at this point the thermal contribution of the element is also removed.

Input File Usage: Use the following option to delete the element from the mesh (default):

*SECTION CONTROLS, ELEMENT DELETION=YES

Keeping the element in the computations

Optionally, you may choose not to remove the element from the mesh, except in the case of three-dimensional beam elements. With element deletion turned off, the overall damage variable is enforced to be $D \leq D_{\max}$. The default value is $D_{\max} = 0.99$ if element deletion is turned off, which ensures that elements will remain active in the simulation with a residual stiffness of at least 1% of the original stiffness. The dimensionality of the stress state of the element affects which stiffness components can become damaged, as discussed below.

In a heat transfer analysis the thermal properties of the material are not affected by damage of the material stiffness.

Input File Usage: Use the following option to keep the element in the computation:

*SECTION CONTROLS, ELEMENT DELETION=NO

Elements with three-dimensional stress states in Abaqus/Explicit

For elements with three-dimensional stress states (including generalized plane strain elements) the shear stiffness will be degraded up to a maximum value, $D_{\max}$, leading to softening of the deviatoric stress components. The bulk stiffness, however, will be degraded only while the material is subjected to negative pressures (i.e., hydrostatic tension); there is no bulk degradation under positive pressures. This corresponds to a fluid-like behavior. Therefore, the degraded deviatoric, $\mathbf{S}$, and pressure, p, stresses are computed as

$$\mathbf{S} = (1 - D_{\text{dev}})\bar{\mathbf{S}},$$
$$p = (1 - D_{\text{vol}})\bar{p},$$

where the deviatoric and volumetric damage variables are given as

$$D_{\text{dev}} = D,$$
$$D_{\text{vol}} = \begin{cases} D & \text{if } \bar{p} \leq 0, \\ 0 & \text{if } \bar{p} > 0. \end{cases}$$

In this case the output variable SDEG contains the value of D_{dev}.

Elements with three-dimensional stress states in Abaqus/Standard

For elements with three-dimensional stress states (including generalized plane strain elements) the stiffness will be degraded uniformly until the maximum degradation, $D_{\max}$, is reached. Output variable SDEG contains the value of D.

Elements with plane stress states

For elements with a plane stress formulation (plane stress, shell, continuum shell, and membrane elements) the stiffness will be degraded uniformly until the maximum degradation, $D_{\max}$, is reached. Output variable SDEG contains the value of D.

Elements with one-dimensional stress states

For elements with a one-dimensional stress state (i.e., truss elements, rebar, and cohesive elements with gasket behavior) their only stress component will be degraded if it is positive (tension). The material stiffness will remain unaffected under compression loading. The stress is, therefore, given by $\sigma = (1 - D_{\mathrm{uni}})\bar{\sigma}$, where the uniaxial damage variable is computed as

$$D_{\mathrm{uni}} = \begin{cases} D & \text{if } \bar{\sigma} \geq 0, \\ 0 & \text{if } \bar{\sigma} < 0. \end{cases}$$

In this case $D_{\max}$ determines the maximum allowed degradation in uniaxial tension ($D \leq D_{\max}$). Output variable SDEG contains the value of D_{uni}.

Convergence difficulties in Abaqus/Standard

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. Some techniques are available in Abaqus/Standard to improve convergence for analyses involving these materials.

Viscous regularization in Abaqus/Standard

You can overcome some of the convergence difficulties associated with softening and stiffness degradation by using the viscous regularization scheme, which causes the tangent stiffness matrix of the softening material to be positive for sufficiently small time increments.

In this regularization scheme a viscous damage variable is defined by the evolution equation:

$$\dot{d}_v = \frac{1}{\eta}(d - d_v),$$

where η is the viscosity coefficient representing the relaxation time of the viscous system and $\boldsymbol{d}$ is the damage variable evaluated in the inviscid base model. The damaged response of the viscous material is computed using the viscous value of the damage variable. Using viscous regularization with a small value of the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime, without compromising results. The basic idea is that the solution of the viscous system relaxes to that of the inviscid case as $t/\eta \to \infty$, where $\boldsymbol{t}$ represents time.

In Abaqus/Standard you can specify the viscous coefficients as part of a section controls definition. For more information, see "Using viscous regularization with cohesive elements, connector elements, and elements that can be used with the damage evolution models for ductile metals and fiber-reinforced composites in Abaqus/Standard" in "Section controls," Section 27.1.4.

Unsymmetric equation solver

In general, if any of the ductile evolution models is used, the material Jacobian matrix will be nonsymmetric. To improve convergence, it is recommended that the unsymmetric equation solver is used in this case.

Using the damage models with rebar

It is possible to use material damage models in elements for which rebar are also defined. The base material contribution to the element stress-carrying capacity diminishes according to the behavior described previously in this section. The rebar contribution to the element stress-carrying capacity will not be affected unless damage is also included in the rebar material definition; in that case the rebar contribution to the element stress-carrying capacity will also be degraded after the damage initiation criterion specified for the rebar is met. For the default choice of element deletion, the element is removed from the mesh when at any one integration location all section points in the base material and rebar are fully degraded.

Elements

Damage evolution for ductile metals can be defined for any element that can be used with the damage initiation criteria for ductile metals in Abaqus (“Damage initiation for ductile metals,” Section 24.2.2).

Output

In addition to the standard output identifiers available in Abaqus (“Abaqus/Standard output variable identifiers,” Section 4.2.1, and “Abaqus/Explicit output variable identifiers,” Section 4.2.2), the following variables have special meaning when damage evolution is specified:

STATUS	Status of element (the status of an element is 1.0 if the element is active, 0.0 if the element is not).
SDEG	Overall scalar stiffness degradation, D.

Additional reference

- Hillerborg, A., M. Modeer, and P. E. Petersson, “Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements,” Cement and Concrete Research, vol. 6, pp. 773–782, 1976.

24.3 Damage and failure for fiber-reinforced composites

- "Damage and failure for fiber-reinforced composites: overview," Section 24.3.1
- "Damage initiation for fiber-reinforced composites," Section 24.3.2
- "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3

24.3.1 DAMAGE AND FAILURE FOR FIBER-REINFORCED COMPOSITES: OVERVIEW

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Progressive damage and failure," Section 24.1.1
- "Damage initiation for fiber-reinforced composites," Section 24.3.2
- "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3
- *DAMAGE INITIATION
- *DAMAGE EVOLUTION
- *DAMAGE STABILIZATION
- "Hashin damage" in "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Abaqus offers a damage model enabling you to predict the onset of damage and to model damage evolution for elastic-brittle materials with anisotropic behavior. The model is primarily intended to be used with fiber-reinforced materials since they typically exhibit such behavior.

This damage model requires specification of the following:

- the undamaged response of the material, which must be linearly elastic (see "Linear elastic behavior," Section 22.2.1);
- a damage initiation criterion (see "Progressive damage and failure," Section 24.1.1, and "Damage initiation for fiber-reinforced composites," Section 24.3.2); and
- a damage evolution response, including a choice of element removal (see "Progressive damage and failure," Section 24.1.1, and "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3).

General concepts of damage in unidirectional lamina

Damage is characterized by the degradation of material stiffness. It plays an important role in the analysis of fiber-reinforced composite materials. Many such materials exhibit elastic-brittle behavior; that is, damage in these materials is initiated without significant plastic deformation. Consequently, plasticity can be neglected when modeling behavior of such materials.

The fibers in the fiber-reinforced material are assumed to be parallel, as depicted in Figure 24.3.1–1. You must specify material properties in a local coordinate system defined by the user. The lamina is in the 1–2 plane, and the local 1 direction corresponds to the fiber direction. You must specify the undamaged material response using one of the methods for defining an orthotropic linear elastic material ("Linear elastic behavior," Section 22.2.1); the most convenient of which is the method for defining an orthotropic material in plane stress ("Defining orthotropic elasticity in plane stress" in "Linear elastic behavior,"

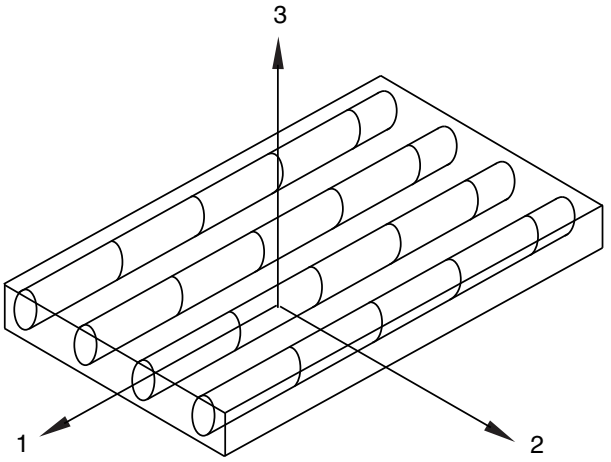


Figure 24.3.1–1 Unidirectional lamina.

Section 22.2.1). However, the material response can also be defined in terms of the engineering constants or by specifying the elastic stiffness matrix directly.

The Abaqus anisotropic damage model is based on the work of Matzenmiller et. al (1995), Hashin and Rotem (1973), Hashin (1980), and Camanho and Davila (2002).

Four different modes of failure are considered:

- fiber rupture in tension;
- fiber buckling and kinking in compression;
- matrix cracking under transverse tension and shearing; and
- matrix crushing under transverse compression and shearing.

In Abaqus the onset of damage is determined by the initiation criteria proposed by Hashin and Rotem (1973) and Hashin (1980), in which the failure surface is expressed in the effective stress space (the stress acting over the area that effectively resists the force). These criteria are discussed in detail in "Damage initiation for fiber-reinforced composites," Section 24.3.2.

The response of the material is computed from

$$\sigma = \mathbf{C_d}\varepsilon,$$

where ε is the strain and $\mathbf{C_d}$ is the elasticity matrix, which reflects any damage and has the form

$$\mathbf{C_d} = \frac{1}{D}\begin{bmatrix} (1-d_f)E_1 & (1-d_f)(1-d_m)\nu_{21}E_1 & 0 \\ (1-d_f)(1-d_m)\nu_{12}E_2 & (1-d_m)E_2 & 0 \\ 0 & 0 & (1-d_s)GD \end{bmatrix},$$

where $D = 1 - (1 - d_f)(1 - d_m)\nu_{12}\nu_{21}$, d_f reflects the current state of fiber damage, d_m reflects the current state of matrix damage, d_s reflects the current state of shear damage, E_1 is the Young's modulus in the fiber direction, E_2 is the Young's modulus in the direction perpendicular to the fibers, G is the shear modulus, and ν_{12} and ν_{21} are Poisson's ratios.

The evolution of the elasticity matrix due to damage is discussed in more detail in "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3; that section also discusses:

- options for treating severe damage ("Maximum degradation and choice of element removal" in "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3); and
- viscous regularization ("Viscous regularization" in "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3).

Elements

The fiber-reinforced composite damage model must be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements.

Additional references

- Camanho, P. P., and C. G. Davila, "Mixed-Mode Decohesion Finite Elements for the Simulation of Delamination in Composite Materials," NASA/TM-2002–211737, pp. 1–37, 2002.
- Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," Journal of Applied Mechanics, vol. 47, pp. 329–334, 1980.
- Hashin, Z., and A. Rotem, "A Fatigue Criterion for Fiber-Reinforced Materials," Journal of Composite Materials, vol. 7, pp. 448–464, 1973.
- Matzenmiller, A., J. Lubliner, and R. L. Taylor, "A Constitutive Model for Anisotropic Damage in Fiber-Composites," Mechanics of Materials, vol. 20, pp. 125–152, 1995.

24.3.2 DAMAGE INITIATION FOR FIBER-REINFORCED COMPOSITES

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Progressive damage and failure," Section 24.1.1
- "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3
- *DAMAGE INITIATION
- "Hashin damage" in "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The material damage initiation capability for fiber-reinforced materials:

- requires that the behavior of the undamaged material is linearly elastic (see "Linear elastic behavior," Section 22.2.1);
- is based on Hashin's theory (Hashin and Rotem, 1973, and Hashin, 1980);
- takes into account four different failure modes: fiber tension, fiber compression, matrix tension, and matrix compression; and
- can be used in combination with the damage evolution model described in "Damage evolution and element removal for fiber-reinforced composites," Section 24.3.3 (see "Failure of blunt notched fiber metal laminates," Section 1.4.6 of the Abaqus Example Problems Manual).

Damage Initiation

Damage initiation refers to the onset of degradation at a material point. In Abaqus the damage initiation criteria for fiber-reinforced composites are based on Hashin's theory (see Hashin and Rotem, 1973, and Hashin, 1980). These criteria consider four different damage initiation mechanisms: fiber tension, fiber compression, matrix tension, and matrix compression.

The initiation criteria have the following general forms:

Fiber tension ($\hat{\sigma}_{11} \geq 0$):

$$F_f^t = (\frac{\hat{\sigma}_{11}}{X^T})^2 + \alpha(\frac{\hat{\tau}_{12}}{S^L})^2.$$

Fiber compression ($\hat{\sigma}_{11} < 0$):

$$F_f^c = (\frac{\hat{\sigma}_{11}}{X^C})^2.$$

Matrix tension ($\hat{\sigma}_{22} \geq 0$):

$$F_m^t = (\frac{\hat{\sigma}_{22}}{Y^T})^2 + (\frac{\hat{\tau}_{12}}{S^L})^2.$$

Matrix compression ($\hat{\sigma_{22}} < 0$):

$$F_m^c = (\frac{\hat{\sigma}_{22}}{2S^T})^2 + [(\frac{Y^C}{2S^T})^2 - 1]\frac{\hat{\sigma}_{22}}{Y^C} + (\frac{\hat{\tau}_{12}}{S^L})^2.$$

In the above equations

X^T	denotes the longitudinal tensile strength;
X^C	denotes the longitudinal compressive strength;
Y^T	denotes the transverse tensile strength;
Y^C	denotes the transverse compressive strength;
S^L	denotes the longitudinal shear strength;
S^T	denotes the transverse shear strength;
α	is a coefficient that determines the contribution of the shear stress to the fiber tensile initiation criterion; and
$\hat{\sigma}_{11}, \hat{\sigma}_{22}, \hat{\tau}_{12}$	are components of the effective stress tensor, $\hat{\sigma}$, that is used to evaluate the initiation criteria and which is computed from:

$$\hat{\sigma} = \mathbf{M}\sigma,$$

where σ is the true stress and $\mathbf{M}$ is the damage operator:

$$\mathbf{M} = \begin{bmatrix} \frac{1}{(1-d_f)} & 0 & 0 \\ 0 & \frac{1}{(1-d_m)} & 0 \\ 0 & 0 & \frac{1}{(1-d_s)} \end{bmatrix}.$$

d_f, d_m, and d_s are internal (damage) variables that characterize fiber, matrix, and shear damage, which are derived from damage variables d_f^t, d_f^c, d_m^t, and d_m^c, corresponding to the four modes previously discussed, as follows:

$$d_f = \begin{cases} d_f^t & \text{if } \hat{\sigma}_{11} \geq 0, \\ d_f^c & \text{if } \hat{\sigma}_{11} < 0, \end{cases}$$
$$d_m = \begin{cases} d_m^t & \text{if } \hat{\sigma}_{22} \geq 0, \\ d_m^c & \text{if } \hat{\sigma}_{22} < 0, \end{cases}$$
$$d_s = 1 - (1 - d_f^t)(1 - d_f^c)(1 - d_m^t)(1 - d_m^c).$$

Prior to any damage initiation and evolution the damage operator, $\mathbf{M}$, is equal to the identity matrix, so $\hat{\sigma} = \sigma$. Once damage initiation and evolution has occurred for at least one mode, the damage operator becomes significant in the criteria for damage initiation of other modes (see "Damage evolution and

element removal for fiber-reinforced composites," Section 24.3.3, for discussion of damage evolution). The effective stress, $\hat{\sigma}$, is intended to represent the stress acting over the damaged area that effectively resists the internal forces.

The initiation criteria presented above can be specialized to obtain the model proposed in Hashin and Rotem (1973) by setting $\alpha = 0.0$ and $S^T = Y^C/2$ or the model proposed in Hashin (1980) by setting $\alpha = 1.0$.

An output variable is associated with each initiation criterion (fiber tension, fiber compression, matrix tension, matrix compression) to indicate whether the criterion has been met. A value of 1.0 or higher indicates that the initiation criterion has been met (see "Output" for further details). If you define a damage initiation model without defining an associated evolution law, the initiation criteria will affect only output. Thus, you can use these criteria to evaluate the propensity of the material to undergo damage without modeling the damage process.

Input File Usage: Use the following option to define the Hashin damage initiation criterion:

*DAMAGE INITIATION, CRITERION=HASHIN, ALPHA=α
X^T, X^C, Y^T, Y^C, S^L, S^T

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Fiber-Reinforced Composites→Hashin Damage**

Elements

The damage initiation criteria must be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1, and, "Abaqus/Explicit output variable identifiers," Section 4.2.2), the following variables relate specifically to damage initiation at a material point in the fiber-reinforced composite damage model:

DMICRT	All damage initiation criteria components.
HSNFTCRT	Maximum value of the fiber tensile initiation criterion experienced during the analysis.
HSNFCCRT	Maximum value of the fiber compressive initiation criterion experienced during the analysis.
HSNMTCRT	Maximum value of the matrix tensile initiation criterion experienced during the analysis.
HSNMCCRT	Maximum value of the matrix compressive initiation criterion experienced during the analysis.

For the variables above that indicate whether an initiation criterion in a damage mode has been satisfied or not, a value that is less than 1.0 indicates that the criterion has not been satisfied, while a value of 1.0 or higher indicates that the criterion has been satisfied. If you define a damage evolution model, the

maximum value of this variable does not exceed 1.0. However, if you do not define a damage evolution model, this variable can have values higher than 1.0, which indicates by how much the criterion has been exceeded.

Additional references

- Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," Journal of Applied Mechanics, vol. 47, pp. 329–334, 1980.
- Hashin, Z., and A. Rotem, "A Fatigue Criterion for Fiber-Reinforced Materials," Journal of Composite Materials, vol. 7, pp. 448–464, 1973.

24.3.3 DAMAGE EVOLUTION AND ELEMENT REMOVAL FOR FIBER-REINFORCED COMPOSITES

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Progressive damage and failure," Section 24.1.1
- "Damage initiation for fiber-reinforced composites," Section 24.3.2
- *DAMAGE EVOLUTION
- "Damage evolution" in "Defining damage," Section 12.9.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The damage evolution capability for fiber-reinforced materials in Abaqus:

- assumes that damage is characterized by progressive degradation of material stiffness, leading to material failure;
- requires linearly elastic behavior of the undamaged material (see "Linear elastic behavior," Section 22.2.1);
- takes into account four different failure modes: fiber tension, fiber compression, matrix tension, and matrix compression;
- uses four damage variables to describe damage for each failure mode;
- must be used in combination with Hashin's damage initiation criteria ("Damage initiation for fiber-reinforced composites," Section 24.3.2);
- is based on energy dissipation during the damage process;
- offers options for what occurs upon failure, including the removal of elements from the mesh; and
- can be used in conjunction with a viscous regularization of the constitutive equations to improve the convergence rate in the softening regime.

Damage evolution

The previous section ("Damage initiation for fiber-reinforced composites," Section 24.3.2) discussed the damage initiation in plane stress fiber-reinforced composites. This section will discuss the post-damage initiation behavior for cases in which a damage evolution model has been specified. Prior to damage initiation the material is linearly elastic, with the stiffness matrix of a plane stress orthotropic material. Thereafter, the response of the material is computed from

$$\sigma = \mathbf{C_d}\varepsilon,$$

where ε is the strain and $\mathbf{C_d}$ is the damaged elasticity matrix, which has the form

$$\mathbf{C_d} = \frac{1}{D}\begin{bmatrix} (1-d_f)E_1 & (1-d_f)(1-d_m)\nu_{21}E_1 & 0 \\ (1-d_f)(1-d_m)\nu_{12}E_2 & (1-d_m)E_2 & 0 \\ 0 & 0 & (1-d_s)GD \end{bmatrix},$$

where $D = 1-(1-d_f)(1-d_m)\nu_{12}\nu_{21}$, d_f reflects the current state of fiber damage, d_m reflects the current state of matrix damage, d_s reflects the current state of shear damage, E_1 is the Young's modulus in the fiber direction, E_2 is the Young's modulus in the matrix direction, G is the shear modulus, and ν_{12} and ν_{21} are Poisson's ratios.

The damage variables d_f, d_m, and d_s are derived from damage variables d_f^t, d_f^c, d_m^t, and d_m^c, corresponding to the four failure modes previously discussed, as follows:

$$d_f = \begin{cases} d_f^t & \text{if } \hat{\sigma}_{11} \geq 0, \\ d_f^c & \text{if } \hat{\sigma}_{11} < 0, \end{cases}$$
$$d_m = \begin{cases} d_m^t & \text{if } \hat{\sigma}_{22} \geq 0, \\ d_m^c & \text{if } \hat{\sigma}_{22} < 0, \end{cases}$$
$$d_s = 1-(1-d_f^t)(1-d_f^c)(1-d_m^t)(1-d_m^c),$$

$\hat{\sigma}_{11}$ and $\hat{\sigma}_{22}$ are components of the effective stress tensor. The effective stress tensor is primarily used to evaluate damage initiation criteria; see "Damage initiation for fiber-reinforced composites," Section 24.3.2, for a description of how the effective stress tensor is computed.

Evolution of damage variables for each mode

To alleviate mesh dependency during material softening, Abaqus introduces a characteristic length into the formulation, so that the constitutive law is expressed as a stress-displacement relation. The damage variable will evolve such that the stress-displacement behaves as shown in Figure 24.3.3–1 in each of the four failure modes. The positive slope of the stress-displacement curve prior to damage initiation corresponds to linear elastic material behavior; the negative slope after damage initiation is achieved by evolution of the respective damage variables according to the equations shown below.

Equivalent displacement and stress for each of the four damage modes are defined as follows:

Fiber tension ($\hat{\sigma}_{11} \geq 0$):

$$\delta_{eq}^{ft} = L^c\sqrt{\langle\varepsilon_{11}\rangle^2 + \alpha\varepsilon_{12}^2},$$

$$\sigma_{eq}^{ft} = \frac{\langle\sigma_{11}\rangle\langle\varepsilon_{11}\rangle + \alpha\tau_{12}\varepsilon_{12}}{\delta_{eq}^{ft}/L^c},$$

Fiber compression ($\hat{\sigma}_{11} < 0$):

$$\delta_{eq}^{fc} = L^c\langle-\varepsilon_{11}\rangle,$$

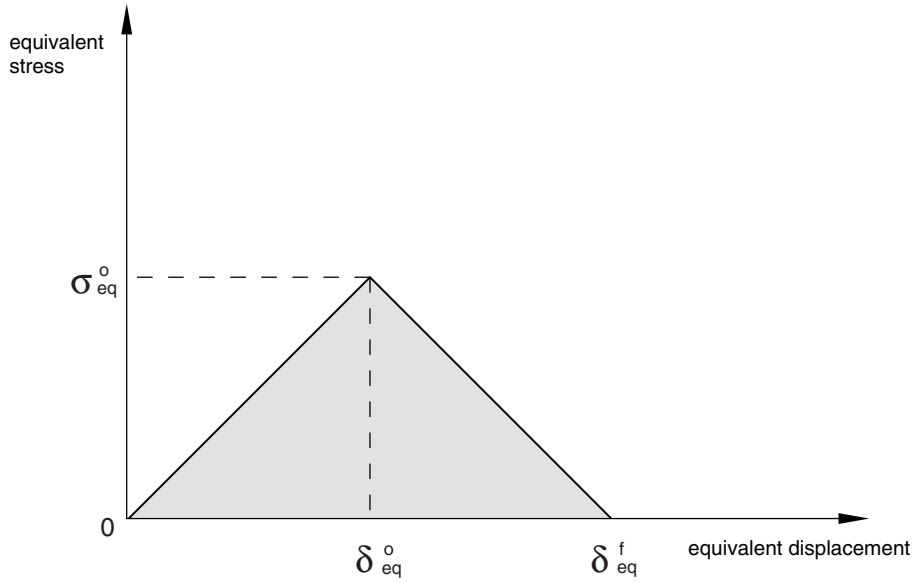


Figure 24.3.3–1 Equivalent stress versus equivalent displacement.

$$\sigma_{eq}^{fc} = \frac{\langle -\sigma_{11} \rangle \langle -\varepsilon_{11} \rangle}{\delta_{eq}^{fc} / L^c}.$$

Matrix tension ($\hat{\sigma}_{22} \geq 0$):

$$\delta_{eq}^{mt} = L^c \sqrt{\langle \varepsilon_{22} \rangle^2 + \varepsilon_{12}^2},$$

$$\sigma_{eq}^{mt} = \frac{\langle \sigma_{22} \rangle \langle \varepsilon_{22} \rangle + \tau_{12} \varepsilon_{12}}{\delta_{eq}^{mt} / L^c}.$$

Matrix compression ($\hat{\sigma}_{22} < 0$):

$$\delta_{eq}^{mc} = L^c \sqrt{\langle -\varepsilon_{22} \rangle^2 + \varepsilon_{12}^2},$$

$$\sigma_{eq}^{mc} = \frac{\langle -\sigma_{22} \rangle \langle -\varepsilon_{22} \rangle + \tau_{12} \varepsilon_{12}}{\delta_{eq}^{mc} / L^c}.$$

The characteristic length, L^c, is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order

element. For membranes and shells it is a characteristic length in the reference surface, computed as the square root of the area. The symbol $\langle\ \rangle$ in the equations above represents the Macaulay bracket operator, which is defined for every $\alpha \in \Re$ as $\langle\alpha\rangle = (\alpha + |\alpha|)/2$.

After damage initiation (i.e., $\delta_{eq} \geq \delta^0_{eq}$) for the behavior shown in Figure 24.3.3–1, the damage variable for a particular mode is given by the following expression

$$d = \frac{\delta^f_{eq}(\delta_{eq} - \delta^0_{eq})}{\delta_{eq}(\delta^f_{eq} - \delta^0_{eq})},$$

where δ^0_{eq} is the initial equivalent displacement at which the initiation criterion for that mode was met and δ^f_{eq} is the displacement at which the material is completely damaged in this failure mode. The above relation is presented graphically in Figure 24.3.3–2.

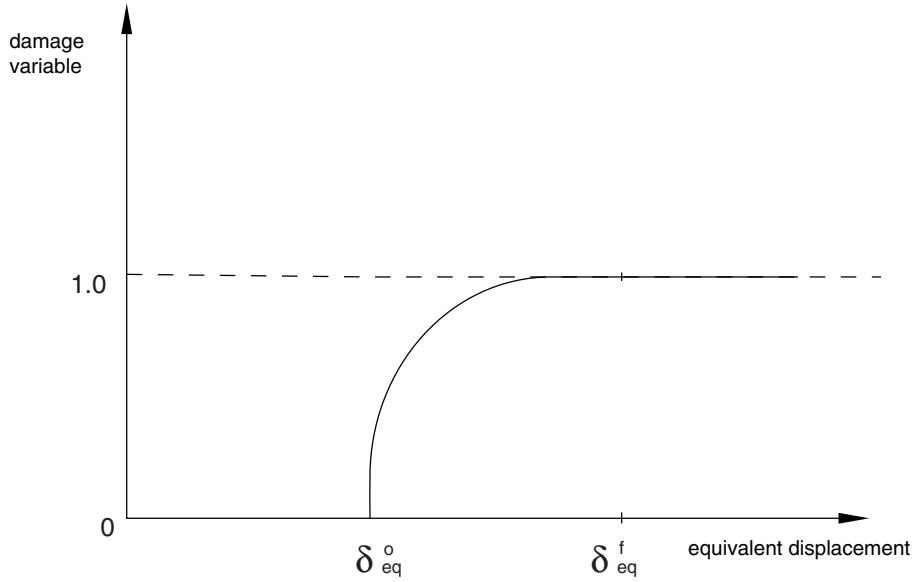


Figure 24.3.3–2 Damage variable as a function of equivalent displacement.

The values of δ^0_{eq} for the various modes depend on the elastic stiffness and the strength parameters specified as part of the damage initiation definition (see "Damage initiation for fiber-reinforced composites," Section 24.3.2). For each failure mode you must specify the energy dissipated due to failure, G^c, which corresponds to the area of the triangle OAC in Figure 24.3.3–3. The values of δ^f_{eq} for the various modes depend on the respective G^c values.

Unloading from a partially damaged state, such as point B in Figure 24.3.3–3, occurs along a linear path toward the origin in the plot of equivalent stress vs. equivalent displacement; this same path is followed back to point B upon reloading as shown in the figure.

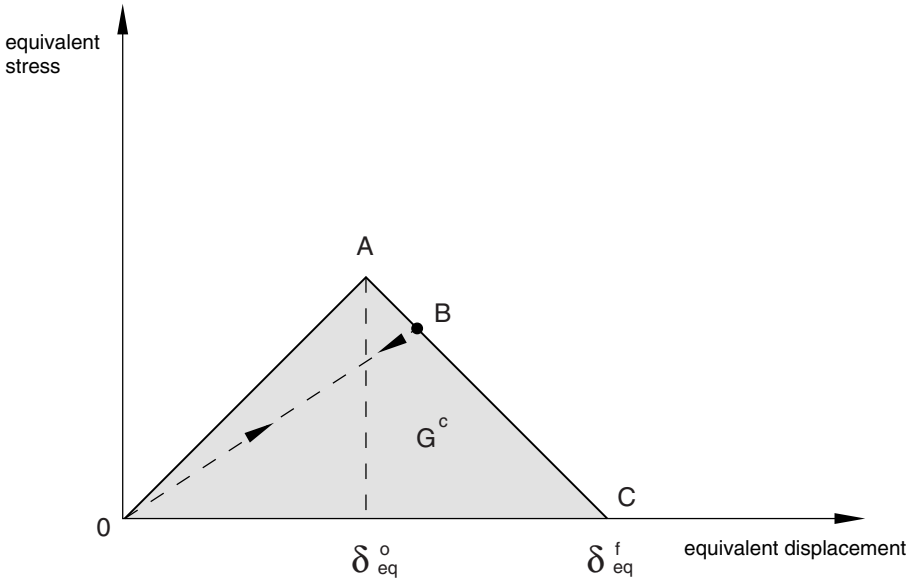


Figure 24.3.3–3 Linear damage evolution.

Input File Usage: Use the following option to define the damage evolution law:

*DAMAGE EVOLUTION, TYPE=ENERGY, SOFTENING=LINEAR
G^c_{ft}, G^c_{fc}, G^c_{mt}, G^c_{mc}

where G^c_{ft}, G^c_{fc}, G^c_{mt}, and G^c_{mc} are energies dissipated during damage for fiber tension, fiber compression, matrix tension, and matrix compression failure modes, respectively.

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damage for Fiber-Reinforced Composites→Hashin Damage**: **Suboptions→Damage Evolution**: **Type: Energy**: **Softening: Linear**

Maximum degradation and choice of element removal

You have control over how Abaqus treats elements with severe damage. By default, the upper bound to all damage variables at a material point is $d_{max} = 1.0$. You can reduce this upper bound as discussed in "Controlling element deletion and maximum degradation for materials with damage evolution" in "Section controls," Section 27.1.4.

By default, in Abaqus/Standard an element is removed (deleted) once damage variables for all failure modes at all material points reach d_{max} (see "Controlling element deletion and maximum degradation for materials with damage evolution" in "Section controls," Section 27.1.4). In Abaqus/Explicit a material point is assumed to fail when either of the damage variables associated with fiber failure modes (tensile or compressive) reaches d_{max} and the element is removed from the mesh

when this condition is satisfied at all of the section points at any one integration location of an element; for example, in the case of shell elements all through-the-thickness section points at any one integration location of the element must fail before the element is removed from the mesh. If an element is removed, the output variable STATUS is set to zero for the element, and it offers no resistance to subsequent deformation. Elements that have been removed are not displayed when you view the deformed model in the Visualization module of Abaqus/CAE (Abaqus/Viewer). However, the elements still remain in the Abaqus model. You can choose to display removed elements by suppressing use of the STATUS variable (see "Selecting the status field output variable," Section 42.5.6 of the Abaqus/CAE User's Manual, in the online HTML version of this manual).

Alternatively, you can specify that an element should remain in the model even after all of the damage variables reach d_{max}. In this case, once all the damage variables reach the maximum value, the stiffness, $\mathbf{C_d}$, remains constant (see the expression for $\mathbf{C_d}$ earlier in this section).

Difficulties associated with element removal in Abaqus/Standard

When elements are removed from the model, their nodes will still remain in the model even if they are not attached to any active elements. When the solution progresses, these nodes might undergo non-physical displacements due to the extrapolation scheme used in Abaqus/Standard to speed up the solution (see "Convergence criteria for nonlinear problems," Section 7.2.3). These non-physical displacements can be prevented by turning off the extrapolation. In addition, applying a point load to a node that is not attached to an active element will cause convergence difficulties since there is no stiffness to resist the load. It is the responsibility of the user to prevent such situations.

Viscous regularization

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. You can overcome some of these convergence difficulties by using the viscous regularization scheme, which causes the tangent stiffness matrix of the softening material to be positive for sufficiently small time increments.

In this regularization scheme a viscous damage variable is defined by the evolution equation:

$$\dot{d}_v = \frac{1}{\eta}(d - d_v),$$

where η is the viscosity coefficient representing the relaxation time of the viscous system and d is the damage variable evaluated in the inviscid backbone model. The damaged response of the viscous material is given as

$$\sigma = \mathbf{C_d}\epsilon,$$

where the damaged elasticity matrix, $\mathbf{C_d}$, is computed using viscous values of damage variables for each failure mode. Using viscous regularization with a small value of the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime, without compromising results. The basic idea is that the solution of the viscous system relaxes to that of the inviscid case as $t/\eta \to \infty$, where t represents time.

Viscous regularization is also available in Abaqus/Explicit. Viscous regularization slows down the rate of increase of damage and leads to increased fracture energy with increasing deformation rates, which can be exploited as an effective method of modeling rate-dependent material behavior.

In Abaqus/Standard the approximate amount of energy associated with viscous regularization over the whole model or over an element set is available using output variable ALLCD.

Defining viscous regularization coefficients

You can specify different values of viscous coefficients for different failure modes.

Input File Usage: Use the following option to define viscous coefficients:

*DAMAGE STABILIZATION

η_{ft}, η_{fc}, η_{mt}, η_{mc}

where η_{ft}, η_{fc}, η_{mt}, η_{mc} are viscosity coefficients for fiber tension, fiber compression, matrix tension, and matrix compression failure modes, respectively.

Abaqus/CAE Usage: Use the following input to define the viscous coefficients for fiber-reinforced materials:

Property module: material editor: **Mechanical→Damage for Fiber-Reinforced Composites→Hashin Damage**: **Suboptions→Damage Stabilization**

Applying a single viscous coefficient in Abaqus/Standard

Alternatively, in Abaqus/Standard you can specify the viscous coefficients as part of a section controls definition. In this case the same viscous coefficient will be applied to all failure modes. For more information, see "Using viscous regularization with cohesive elements, connector elements, and elements that can be used with the damage evolution models for ductile metals and fiber-reinforced composites in Abaqus/Standard" in "Section controls," Section 27.1.4.

Material damping

If stiffness proportional damping is specified in combination with the damage evolution law for fiber-reinforced materials, Abaqus calculates the damping stresses using the damaged elastic stiffness.

Elements

The damage evolution law for fiber-reinforced materials must be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements.

Output

In addition to the standard output identifiers available in Abaqus ("Abaqus/Standard output variable identifiers," Section 4.2.1), the following variables relate specifically to damage evolution in the fiber-reinforced composite damage model:

STATUS	Status of the element (the status of an element is 1.0 if the element is active, 0.0 if the element is not). The value of this variable is set to 0.0 only if damage has occurred in all the damage modes.
DAMAGEFT	Fiber tensile damage variable.
DAMAGEFC	Fiber compressive damage variable.
DAMAGEMT	Matrix tensile damage variable.
DAMAGEMC	Matrix compressive damage variable.
DAMAGESHR	Shear damage variable.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
ELDMD	Total energy dissipated in the element by damage.
DMENER	Energy dissipated per unit volume by damage.
ALLDMD	Energy dissipated in the whole (or partial) model by damage.
ECDDEN	Energy per unit volume in the element that is associated with viscous regularization.
ELCD	Total energy in the element that is associated with viscous regularization.
CENER	Energy per unit volume that is associated with viscous regularization.
ALLCD	The approximate amount of energy over the whole model or over an element set that is associated with viscous regularization.

24.4 Damage and failure for ductile materials in low-cycle fatigue analysis

- "Damage and failure for ductile materials in low-cycle fatigue analysis: overview," Section 24.4.1
- "Damage initiation for ductile materials in low-cycle fatigue," Section 24.4.2
- "Damage evolution for ductile materials in low-cycle fatigue," Section 24.4.3

24.4.1 DAMAGE AND FAILURE FOR DUCTILE MATERIALS IN LOW-CYCLE FATIGUE ANALYSIS: OVERVIEW

Product: Abaqus/Standard

References

- "Progressive damage and failure," Section 24.1.1
- "Damage initiation for ductile materials in low-cycle fatigue," Section 24.4.2
- "Damage evolution for ductile materials in low-cycle fatigue," Section 24.4.3
- "Low-cycle fatigue analysis using the direct cyclic approach," Section 6.2.7
- *DAMAGE INITIATION
- *DAMAGE EVOLUTION

Overview

Abaqus/Standard offers a general capability for modeling progressive damage and failure of ductile materials due to stress reversals and the accumulation of inelastic strain energy in a low-cycle fatigue analysis using the direct cyclic approach. In the most general case this requires the specification of the following:

- the undamaged ductile materials in any elements (including cohesive elements based on a continuum approach) whose response is defined in terms of a continuum-based constitutive model ("Material library: overview," Section 21.1.1);
- a damage initiation criterion ("Damage initiation for ductile materials in low-cycle fatigue," Section 24.4.2); and
- a damage evolution response ("Damage evolution for ductile materials in low-cycle fatigue," Section 24.4.3).

A summary of the general framework for progressive damage and failure in Abaqus is given in "Progressive damage and failure," Section 24.1.1. This section provides an overview of the damage initiation criteria and damage evolution law for ductile materials in a low-cycle fatigue analysis using the direct cyclic approach.

General concepts of damage of ductile materials in low-cycle fatigue

Accurately and effectively predicting the fatigue life for an inelastic structure, such as a solder joint in an electronic chip packaging, subjected to sub-critical cyclic loading is a challenging problem. Cyclic thermal or mechanical loading often leads to stress reversals and the accumulation of inelastic strain, which may in turn lead to the initiation and propagation of a crack. The low-cycle fatigue analysis capability in Abaqus/Standard uses a direct cyclic approach ("Low-cycle fatigue analysis using the direct cyclic approach," Section 6.2.7) to model progressive damage and failure based on a continuum

damage approach. The damage initiation ("Damage initiation for ductile materials in low-cycle fatigue," Section 24.4.2) and evolution ("Damage evolution for ductile materials in low-cycle fatigue," Section 24.4.3) are characterized by the stabilized accumulated inelastic hysteresis strain energy per cycle proposed by Darveaux (2002) and Lau (2002).

The damage evolution law describes the rate of degradation of the material stiffness per cycle once the corresponding initiation criterion has been reached. For damage in ductile materials Abaqus/Standard assumes that the degradation of the stiffness can be modeled using a scalar damage variable, D. At any given cycle during the analysis the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\bar{\boldsymbol{\sigma}},$$

where $\bar{\boldsymbol{\sigma}}$ is the effective (or undamaged) stress tensor that would exist in the material in the absence of damage computed in the current increment. The material has lost its load carrying capacity when $D = 1$.

Elements

The failure modeling capability for ductile materials can be used with any elements (including cohesive elements based on a continuum approach) in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom).

Additional references

- Darveaux, R., "Effect of Simulation Methodology on Solder Joint Crack Growth Correlation and Fatigue Life Prediction," Journal of Electronic Packaging, vol. 124, pp. 147–154, 2002.
- Lau, J., S. Pan, and C. Chang, "A New Thermal-Fatigue Life Prediction Model for Wafer Level Chip Scale Package (WLCSP) Solder Joints," Journal of Electronic Packaging, vol. 124, pp. 212–220, 2002.

24.4.2 DAMAGE INITIATION FOR DUCTILE MATERIALS IN LOW-CYCLE FATIGUE

Product: Abaqus/Standard

References

- "Progressive damage and failure," Section 24.1.1
- ∗DAMAGE INITIATION

Overview

The material damage initiation capability for ductile materials based on inelastic hysteresis energy:

- is intended as a general capability for predicting initiation of damage in ductile materials in a low-cycle fatigue analysis;
- can be used in combination with the damage evolution law for ductile materials described in "Damage evolution for ductile materials in low-cycle fatigue," Section 24.4.3; and
- can be used only in a low-cycle fatigue analysis using the direct cyclic approach ("Low-cycle fatigue analysis using the direct cyclic approach," Section 6.2.7).

Damage initiation criteria for ductile materials

The damage initiation criterion is a phenomenological model for predicting the onset of damage due to stress reversals and the accumulation of inelastic strain in a low-cycle fatigue analysis. It is characterized by the accumulated inelastic hysteresis energy per cycle, Δw, in a material point when the structure response is stabilized in the cycle. The cycle number in which damage is initiated is given by

$$N_0 = c_1 \Delta w^{c_2},$$

where c_1 and c_2 are material constants. The value of c_1 is dependent on the system of units in which you are working; some care is required to modify c_1 when converting to a different system of units.

The initiation criterion can be used in conjunction with any ductile material.

Input File Usage: ∗DAMAGE INITIATION, CRITERION=HYSTERESIS ENERGY

Elements

The damage initiation criteria for ductile materials can be used with any elements in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom). This includes cohesive elements based on a continuum approach ("Modeling of an adhesive layer of finite thickness" in "Defining the constitutive response of cohesive elements using a continuum approach," Section 32.5.5).

Output

In addition to the standard output identifiers available in Abaqus/Standard ("Abaqus/Standard output variable identifiers," Section 4.2.1), the following variable has special meaning when a damage initiation criterion is specified:

CYCLEINI	Number of cycles to initialize the damage at the material point.

24.4.3 DAMAGE EVOLUTION FOR DUCTILE MATERIALS IN LOW-CYCLE FATIGUE

Product: Abaqus/Standard

References

- "Progressive damage and failure," Section 24.1.1
- *DAMAGE EVOLUTION

Overview

The damage evolution capability for ductile materials based on inelastic hysteresis energy:

- assumes that damage is characterized by the progressive degradation of the material stiffness, leading to material failure;
- must be used in combination with a damage initiation criterion for ductile materials in low-cycle fatigue analysis ("Damage initiation for ductile materials in low-cycle fatigue," Section 24.4.2);
- uses the inelastic hysteresis energy per stabilized cycle to drive the evolution of damage after damage initiation; and
- must be used in conjunction with the linear elastic material model ("Linear elastic behavior," Section 22.2.1), the porous elastic material model ("Elastic behavior of porous materials," Section 22.3.1), or the hypoelastic material model ("Hypoelastic behavior," Section 22.4.1).

Damage evolution based on accumulated inelastic hysteresis energy

Once the damage initiation criterion ("Damage initiation for ductile materials in low-cycle fatigue," Section 24.4.2) is satisfied at a material point, the damage state is calculated and updated based on the inelastic hysteresis energy for the stabilized cycle. The rate of the damage in a material point per cycle is given by

$$\frac{dD}{dN} = \frac{c_3 \Delta w^{c_4}}{L},$$

where c_3 and c_4 are material constants, and L is the characteristic length associated with an integration point. The value of c_3 is dependent on the system of units in which you are working; some care is required to modify c_3 when converting to a different system of units.

For damage in ductile materials Abaqus/Standard assumes that the degradation of the elastic stiffness can be modeled using the scalar damage variable, D. At any given loading cycle during the analysis the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\bar{\boldsymbol{\sigma}},$$

where $\bar{\sigma}$ is the effective (or undamaged) stress tensor that would exist in the material in the absence of damage computed in the current increment. The material has completely lost its load carrying capacity when $D = 1$. You can remove the element from the mesh if all of the section points at all integration locations have lost their loading carrying capability.

Input File Usage: *DAMAGE EVOLUTION, TYPE=HYSTERESIS ENERGY

Mesh dependency and characteristic length

The implementation of the damage evolution model requires the definition of a characteristic length associated with an integration point. The characteristic length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic length is used because the direction in which fracture occurs is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which the damage occurs: some mesh sensitivity remains because of this effect, and elements that are as close to square as possible are recommended. However, since the damage evolution law is energy based, mesh dependency of the results may be alleviated.

Maximum degradation and element removal

You can control how Abaqus/Standard treats elements with severe damage.

Defining the upper bound to the damage variable

By default, the upper bound to all damage variables at a material point is $D_{max} = 1.0$. You can reduce this upper bound as discussed in "Controlling element deletion and maximum degradation for materials with damage evolution" in "Section controls," Section 27.1.4.

Input File Usage: *SECTION CONTROLS, MAX DEGRADATION=$D_{\max}$

Controlling element removal for damaged elements

By default, in Abaqus/Standard an element is removed (deleted) once D reaches $D_{\max}$ at all of the section points at all integration locations in the element. If an element is removed, the output variable STATUS is set to zero for the element, and it offers no resistance to subsequent deformation. However, the element still remains in the Abaqus/Standard model and may be visible during postprocessing. In the Visualization module of Abaqus/CAE, you can suppress the display of elements based on their status (see "Selecting the status field output variable," Section 42.5.6 of the Abaqus/CAE User's Manual, in the online HTML version of this manual).

Alternatively, you can specify that an element should remain in the model even after all of the damage variables reach D_{max}. In this case, once all the damage variables reach the maximum value, the stiffness remains constant.

Input File Usage: Use the following option to delete failed elements from the mesh (default):

*SECTION CONTROLS, ELEMENT DELETION=YES

Use the following option to keep failed elements in the mesh computations:

*SECTION CONTROLS, ELEMENT DELETION=NO

Difficulties associated with element removal in Abaqus/Standard

When elements are removed from the model, their nodes remain in the model even if they are not attached to any active elements. When the solution progresses, these nodes might undergo non-physical displacements in Abaqus/Standard. In addition, applying a point load to a node that is not attached to an active element will cause convergence difficulties since there is no stiffness to resist the load. It is the responsibility of the user to prevent such situations.

Elements

Damage evolution for ductile materials can be defined for any element that can be used with the damage initiation criteria for a low-cycle fatigue analysis in Abaqus/Standard (“Damage initiation for ductile materials in low-cycle fatigue,” Section 24.4.2).

Output

In addition to the standard output identifiers available in Abaqus/Standard (“Abaqus/Standard output variable identifiers,” Section 4.2.1), the following variables have special meaning when damage evolution is specified:

STATUS	Status of element (the status of an element is 1.0 if the element is active, 0.0 if the element is not).
SDEG	Overall scalar stiffness degradation, D.

25. Hydrodynamic Properties

Overview 25.1
Equations of state 25.2

25.1 Overview

- “Hydrodynamic behavior: overview,” Section 25.1.1

25.1.1 HYDRODYNAMIC BEHAVIOR: OVERVIEW

The material library in Abaqus/Explicit includes several equation of state models to describe the hydrodynamic behavior of materials. An equation of state is a constitutive equation that defines the pressure as a function of the density and the internal energy ("Equation of state," Section 25.2.1). The following equations of state are supported in Abaqus/Explicit:

- **Mie-Grüneisen equation of state:** The Mie-Grüneisen equation of state ("Mie-Grüneisen equations of state" in "Equation of state," Section 25.2.1) is used to model materials at high pressure. It is linear in energy and assumes a linear relationship between the shock velocity and the particle velocity.
- **Tabulated equation of state:** The tabulated equation of state ("Tabulated equation of state" in "Equation of state," Section 25.2.1) is used to model the hydrodynamic response of materials that exhibit sharp transitions in the pressure-density relationship, such as those induced by phase transformations. It is linear in energy.
- $P-\alpha$ **equation of state:** The $P-\alpha$ equation of state ("$P-\alpha$ equation of state" in "Equation of state," Section 25.2.1) is designed for modeling the compaction of ductile porous materials. The constitutive model captures the irreversible compaction behavior at low stresses and predicts the correct thermodynamic behavior at high pressures for the fully compacted solid material. It is used in combination with either the Mie-Grüneisen equation of state or the tabulated equation of state to describe the solid phase.
- **JWL high explosive equation of state:** The Jones-Wilkens-Lee (or JWL) equation of state ("JWL high explosive equation of state" in "Equation of state," Section 25.2.1) models the pressure generated by the release of chemical energy in an explosive. This model is implemented in a form referred to as a programmed burn, which means that the reaction and initiation of the explosive is not determined by shock in the material. Instead, the initiation time is determined by a geometric construction using the detonation wave speed and the distance of the material point from the detonation points.
- **Ideal gas equation of state:** The ideal gas equation of state ("Ideal gas equation of state" in "Equation of state," Section 25.2.1) is an idealization to real gas behavior and can be used to model any gases approximately under appropriate conditions (e.g., low pressure and high temperature).

Deviatoric behavior

The material modeled by an equation of state may have no deviatoric strength or may have either isotropic elastic or viscous (both Newtonian and non-Newtonian) deviatoric behavior ("Deviatoric behavior" in "Equation of state," Section 25.2.1). The elastic model can be used by itself or in conjunction with the Mises, the Johnson-Cook, or the extended Drucker-Prager plasticity models to model hydrodynamic materials with elastic-plastic deviatoric behavior.

Thermal strain

Thermal expansion cannot be introduced for any of the equation of state models.

25.2 Equations of state

- “Equation of state,” Section 25.2.1

25.2.1 EQUATION OF STATE

Products: Abaqus/Explicit Abaqus/CAE

References

- "Hydrodynamic behavior: overview," Section 25.1.1
- "Material library: overview," Section 21.1.1
- *EOS
- *EOS COMPACTION
- *ELASTIC
- *VISCOSITY
- *DETONATION POINT
- *GAS SPECIFIC HEAT
- *REACTION RATE
- *TENSILE FAILURE
- "Defining equations of state" in "Defining other mechanical models," Section 12.9.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Equations of state:

- provide a hydrodynamic material model in which the material's volumetric strength is determined by an equation of state;
- determine the pressure (positive in compression) as a function of the density, ρ, and the specific energy (the internal energy per unit mass), E_m: $p = f(\rho, E_m)$;
- are available as Mie-Grüneisen equations of state (thus providing the linear $U_s - U_p$ Hugoniot form);
- are available as tabulated equations of state linear in energy;
- are available as $P - \alpha$ equations of state for the compaction of ductile porous materials and must be used in conjunction with either the Mie-Grüneisen or the tabulated equation of state for the solid phase;
- are available as JWL high explosive equations of state;
- are available as ignition and growth equations of state;
- are available in the form of an ideal gas;
- assume an adiabatic condition unless a dynamic fully coupled temperature-displacement analysis is used;
- can be used to model a material that has only volumetric strength (the material is assumed to have no shear strength) or a material that also has isotropic elastic or viscous deviatoric behavior;

- can be used with the Mises ("Classical metal plasticity," Section 23.2.1) or the Johnson-Cook ("Johnson-Cook plasticity," Section 23.2.7) plasticity models;
- can be used with the extended Drucker-Prager ("Extended Drucker-Prager models," Section 23.3.1) plasticity models (without plastic dilation); and
- can be used with the tensile failure model ("Dynamic failure models," Section 23.2.8) to model dynamic spall or a pressure cutoff.

Energy equation and Hugoniot curve

The equation for conservation of energy equates the increase in internal energy per unit mass, E_m, to the rate at which work is being done by the stresses and the rate at which heat is being added. In the absence of heat conduction the energy equation can be written as

$$\rho \frac{\partial E_m}{\partial t} = (p - p_{bv}) \frac{1}{\rho} \frac{\partial \rho}{\partial t} + \mathbf{S} : \dot{\mathbf{e}} + \rho \dot{Q},$$

where p is the pressure stress defined as positive in compression, p_{bv} is the pressure stress due to the bulk viscosity, $\mathbf{S}$ is the deviatoric stress tensor, $\dot{\mathbf{e}}$ is the deviatoric part of strain rate, and $\dot{Q}$ is the heat rate per unit mass.

The equation of state is assumed for the pressure as a function of the current density, ρ, and the internal energy per unit mass, E_m:

$$p = f(\rho, E_m),$$

which defines all the equilibrium states that can exist in a material. The internal energy can be eliminated from the above equation to obtain a p versus V relationship (where V is the current volume) or, equivalently, a p versus $1/\rho$ relationship that is unique to the material described by the equation of state model. This unique relationship is called the Hugoniot curve and is the locus of p–V states achievable behind a shock (see Figure 25.2.1–1).

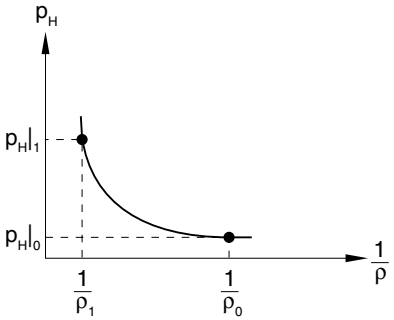


Figure 25.2.1–1 A schematic representation of a Hugoniot curve.

The Hugoniot pressure, p_H, is a function of density only and can be defined, in general, from fitting experimental data.

An equation of state is said to be linear in energy when it can be written in the form

$$p = f + gE_m,$$

where $f(\rho)$ and $g(\rho)$ are functions of density only and depend on the particular equation of state model.

Mie-Grüneisen equations of state

A Mie-Grüneisen equation of state is linear in energy. The most common form is

$$p - p_H = \Gamma\rho(E_m - E_H),$$

where p_H and E_H are the Hugoniot pressure and specific energy (per unit mass) and are functions of density only, and Γ is the Grüneisen ratio defined as

$$\Gamma = \Gamma_0 \frac{\rho_0}{\rho},$$

where Γ_0 is a material constant and ρ_0 is the reference density.

The Hugoniot energy, E_H, is related to the Hugoniot pressure by

$$E_H = \frac{p_H \eta}{2\rho_0},$$

where $\eta = 1 - \rho_0/\rho$ is the nominal volumetric compressive strain. Elimination of Γ and E_H from the above equations yields

$$p = p_H \left(1 - \frac{\Gamma_0 \eta}{2}\right) + \Gamma_0 \rho_0 E_m.$$

The equation of state and the energy equation represent coupled equations for pressure and internal energy. Abaqus/Explicit solves these equations simultaneously at each material point.

Linear $U_s - U_p$ Hugoniot form

A common fit to the Hugoniot data is given by

$$p_H = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2},$$

where c_0 and s define the linear relationship between the linear shock velocity, U_s, and the particle velocity, U_p, as follows:

$$U_s = c_0 + sU_p.$$

With the above assumptions the linear $U_s - U_p$ Hugoniot form is written as

$$p = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2}(1 - \frac{\Gamma_0 \eta}{2}) + \Gamma_0 \rho_0 E_m,$$

where $\rho_0 c_0^2$ is equivalent to the elastic bulk modulus at small nominal strains.

There is a limiting compression given by the denominator of this form of the equation of state

$$\eta_{lim} = \frac{1}{s}$$

or

$$\rho_{lim} = \frac{s\rho_0}{s - 1}\ .$$

At this limit there is a tensile minimum; thereafter, negative sound speeds are calculated for the material.

Input File Usage: Use both of the following options:

∗DENSITY *(to specify the reference density ρ_0)*
∗EOS, TYPE=USUP *(to specify the variables c_0, **s**, and Γ_0)*

Abaqus/CAE Usage: Property module: material editor:
General→Density *(to specify the reference density ρ_0)*
Mechanical→Eos: **Type: Us - Up** *(to specify the variables c_0, **s**, and Γ_0)*

Initial state

The initial state of the material is determined by the initial values of specific energy, E_m, and pressure stress, ***p***. Abaqus/Explicit will automatically compute the initial density, ρ, that satisfies the equation of state, $p = f(\rho, E_m)$. You can define the initial specific energy and initial stress state (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). The initial pressure used by the equation of state is inferred from the specified stress states. If no initial conditions are specified, Abaqus/Explicit will assume that the material is at its reference state:

$$E_m = 0,$$
$$p = 0,$$
$$\rho = \rho_0.$$

Input File Usage: Use either or both of the following options, as required:

∗INITIAL CONDITIONS, TYPE=SPECIFIC ENERGY
∗INITIAL CONDITIONS, TYPE=STRESS

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

Tabulated equation of state

The tabulated equation of state provides flexibility in modeling the hydrodynamic response of materials that exhibit sharp transitions in the pressure-density relationship, such as those induced by phase transformations. The tabulated equation of state is linear in energy and assumes the form

$$p = f_1(\varepsilon_{\text{vol}}) + \rho_0 f_2(\varepsilon_{\text{vol}}) E_m,$$

where $f_1(\varepsilon_{\text{vol}})$ and $f_2(\varepsilon_{\text{vol}})$ are functions of the logarithmic volumetric strain ε_{vol} only, with $\varepsilon_{\text{vol}} = \ln(\rho_0/\rho)$, and ρ_0 is the reference density.

You can specify the functions $f_1(\varepsilon_{\text{vol}})$ and $f_2(\varepsilon_{\text{vol}})$ directly in tabular form. The tabular entries must be given in descending values of the volumetric strain (that is, from the most tensile to the most compressive states). Abaqus/Explicit will use a piecewise linear relationship between data points. Outside the range of specified values of volumetric strains, the functions are extrapolated based on the last slope computed from the data.

Input File Usage: Use both of the following options:

*DENSITY *(to specify the reference density ρ_0)*
*EOS, TYPE=TABULAR *(to specify f_1 and f_2 as functions of ε_{vol})*

Abaqus/CAE Usage: The tabulated equation of state is not supported in Abaqus/CAE.

Initial state

The initial state of the material is determined by the initial values of specific energy, E_m, and pressure stress, p. Abaqus/Explicit automatically computes the initial density, ρ, that satisfies the equation of state. You can define the initial specific energy and initial stress state (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). The initial pressure used by the equation of state is inferred from the specified stress states. If no initial conditions are specified, Abaqus/Explicit assumes that the material is at its reference state:

$$\begin{aligned} E_m &= 0, \\ p &= 0, \\ \rho &= \rho_0 \quad (\varepsilon_{\text{vol}} = 0). \end{aligned}$$

Input File Usage: Use either or both of the following options, as required:

*INITIAL CONDITIONS, TYPE=SPECIFIC ENERGY
*INITIAL CONDITIONS, TYPE=STRESS

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

$P-\alpha$ equation of state

The $P-\alpha$ equation of state is designed for modeling the compaction of ductile porous materials. The implementation in Abaqus/Explicit is based on the model proposed by Hermann (1968) and Carroll and Holt (1972). The constitutive model provides a detailed description of the irreversible compaction behavior at low stresses and predicts the correct thermodynamic behavior at high pressures for the fully compacted solid material. In Abaqus/Explicit the solid phase is assumed to be governed by either the Mie-Grüneisen equation of state or the tabulated equation of state. The relevant properties of the porous material in the virgin state, to be discussed later, and the material properties of the solid phase are specified separately.

The porosity of the material, n, is defined as the ratio of pore volume, V_p, to total volume, $V = V_s + V_p$, where V_s is the solid volume. The porosity remains in the range $0 \leq n < 1$, with 0 indicating full compaction. It is convenient to introduce a scalar variable α, sometimes referred to as "distension," defined as the ratio of the density of the solid material, ρ_s, to the density of the porous material, ρ, both evaluated at the same temperature and pressure:

$$\alpha = \frac{\rho_s}{\rho} \geq 1.$$

For a fully compacted material $\alpha = 1$; otherwise, α is greater than 1. Assuming that the density of the pores is negligible compared to that of the solid phase, α can be expressed in terms of the porosity n as

$$\alpha = \frac{\rho_s}{\rho} = \frac{V}{V_s} = \frac{V}{V - V_p} = \frac{1}{1 - V_p/V} = \frac{1}{1-n}.$$

An equation of state is assumed for the pressure of the porous material as a function of α; current density, ρ; and internal energy per unit mass, E_m, in the form

$$p = p(\alpha, \rho, E_m).$$

Assuming that the pores carry no pressure, it follows from equilibrium considerations that when a pressure p is applied to the porous material, it gives rise to a volume-average pressure in the solid phase equal to $p_s = \alpha p$. Assuming that the specific internal energies of the porous material and the solid matrix are the same (i.e., neglecting the surface energy of the pores), the equation of state of the porous material can be expressed as

$$p(\alpha, \rho, E_m) = \frac{1}{\alpha} p_s(\alpha\rho, E_m) = \frac{1}{\alpha} p_s(\rho_s, E_m),$$

where $p_s(\rho_s, E_m)$ is the equation of state of the solid material. For the fully compacted material (that is, when $\alpha = 1$), the $P-\alpha$ equation of state reduces to that of the solid phase, therefore predicting the correct thermodynamic behavior at high pressures.

The $P-\alpha$ equation of state must be supplemented by an equation that describes the behavior of α as a function of the thermodynamic state. This equation takes the form

$$\alpha = A(p, \alpha_{min}),$$

where α_{min} is a state variable corresponding to the minimum value attained by α during plastic (irreversible) compaction of the material. The state variable is initialized to the elastic limit α_e for a material that is at its virgin state. The specific form of the function $A(p, \alpha_{min})$ used by Abaqus/Explicit is illustrated in Figure 25.2.1–2 and is discussed next.

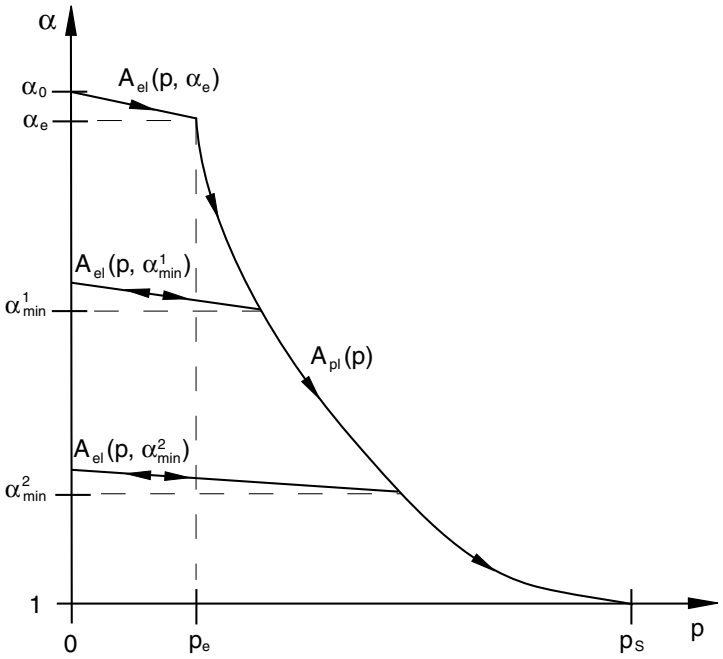


Figure 25.2.1–2 $P - \alpha$ elastic and plastic curves for the description of compaction of ductile porous materials.

The function $A(p, \alpha_{min})$ captures the general behavior to be expected in a ductile porous material. The unloaded virgin state corresponds to the value $\alpha_0 = 1/(1 - n_0)$, where n_0 is the reference porosity of the material. Initial compression of the porous material is assumed to be elastic. Recall that decreasing porosity corresponds to a reduction in α. As the pressure increases beyond the elastic limit, p_e, the pores in the material start to crush, leading to irreversible compaction and permanent (plastic) volume change. Unloading from a partially compacted state follows a new elastic curve that depends on the maximum compaction (or, alternatively, the minimum value of α) ever attained during the deformation history of the material. The absolute value of the slope of the elastic curve decreases as α_{min} decreases, as will be quantified later. The material becomes fully compacted when the pressure reaches the compaction pressure p_S; at that point $\alpha = \alpha_{min} = 1$, a value that is retained forever. The function $A(p, \alpha_{min})$

therefore has multiple branches: a plastic branch, $A_{pl}(p)$, and multiple elastic branches, $A_{el}(p, \alpha_{min})$, corresponding to elastic unloading from partially compacted states. The appropriate branch of $\boldsymbol{A}$ is selected according to the following rule:

$$\alpha = A(p, \alpha_{min}) = \begin{cases} A_{pl}(p) & \text{if} \quad A_{pl}(p) \le \alpha_{min} \\ A_{el}(p, \alpha_{min}) & \text{if} \quad A_{pl}(p) > \alpha_{min} \end{cases}$$

These expressions can be inverted to solve for $\boldsymbol{p}$:

$$p = P(\alpha, \alpha_{min}) = \begin{cases} P_{pl}(\alpha) & \text{if} \quad \alpha \le \alpha_{min} \\ P_{el}(\alpha, \alpha_{min}) & \text{if} \quad \alpha > \alpha_{min} \end{cases}$$

The equation for the plastic curve takes the form

$$A_{pl}(p) = 1 + (\alpha_e - 1)\left(\frac{p_S - p}{p_S - p_e}\right)^2$$

or, alternatively,

$$P_{pl}(\alpha) = p_S - (p_S - p_e)\left(\frac{\alpha - 1}{\alpha_e - 1}\right)^{\frac{1}{2}}.$$

The elastic curve originally proposed by Hermann (1968) is given by the differential equation

$$\frac{d\bar{A}_{el}}{dp}(\alpha) = \frac{\alpha^2}{K_0}\left(1 - \frac{1}{h^2(\alpha)}\right), \qquad h(\alpha) = 1 + \frac{(c_e - c_s)(\alpha - 1)}{c_s(\alpha_0 - 1)},$$

where $K_0 = \rho_{s0} c_s^2$ is the elastic bulk modulus of the solid material at small nominal strains; ρ_{s0} is the reference density of the solid; and c_s and c_e are the reference sound speeds in the solid and virgin (porous) materials, respectively.

If the solid phase is modeled using the Mie-Grüneisen equation of state, c_s is given directly by the reference sound speed, c_0. On the other hand, if the solid phase is modeled using the tabulated equation of state, c_s is computed from the initial bulk modulus and reference density of the solid material, $c_s = \sqrt{K_0/\rho_{s0}}$. In this case the reference density is required to be constant; it cannot be a function of temperature or field variables.

Following Wardlaw et al. (1996), the above equation for the elastic curve in Abaqus/Explicit is simplified and replaced by the linear relations

$$A_{el}(p, \alpha_{min}) = \alpha_{min} + (p - P_{pl}(\alpha_{min}))\left.\frac{d\bar{A}_{el}}{dp}\right|_{\alpha=\alpha_{min}}$$

and

$$P_{el}(\alpha, \alpha_{min}) = P_{pl}(\alpha_{min}) + \frac{(\alpha - \alpha_{min})}{\left.\frac{d\bar{A}_{el}}{dp}\right|_{\alpha=\alpha_{min}}}.$$

Input File Usage: Use the following option to specify the reference density of the solid phase, ρ_{s0}:

*DENSITY

Use one of the following two options to specify additional material properties for the solid phase:

*EOS, TYPE=USUP *(if the solid phase is modeled using the Mie-Grüneisen equation of state)*
*EOS, TYPE=TABULAR *(if the solid phase is modeled using the tabulated equation of state)*

Use the following option to specify the properties of the porous material (the reference sound speed, c_e; the reference porosity, n_0; the elastic limit, p_e; and the compaction pressure, p_S):

*EOS COMPACTION

Abaqus/CAE Usage: Only the Mie-Grüneisen equation of state is supported for the solid phase in Abaqus/CAE.

Property module: material editor:
General→Density *(to specify the reference density ρ_0)*
Mechanical→Eos: **Type: Us - Up** *(to specify the variables c_0, s, and Γ_0)*
Mechanical→Eos: **Suboptions→ Eos Compaction** *(to specify the reference sound speed, c_e; the porosity of the unloaded material, n_0; the pressure required to initialize plastic behavior, p_e; and the pressure at which all pores are crushed, p_S)*

Initial state

The initial state of the porous material is determined from the initial values of porosity, $n = (\alpha - 1)/\alpha$; specific energy, E_m; and pressure stress, p. Abaqus/Explicit automatically computes the initial density, ρ, that satisfies the equation of state, $p = f(\alpha, \rho, E_m)$. You can define the initial porosity, initial specific energy, and initial stress state (see "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). If no initial conditions are given, Abaqus/Explicit assumes that the material is at its virgin state:

$$\begin{aligned} E_m &= 0, \\ p &= 0, \\ \alpha &= \alpha_0, \quad (n = n_0), \\ \rho &= \rho_{s0}/\alpha_0. \end{aligned}$$

Abaqus/Explicit will issue an error message if the initial $p-\alpha$ state lies outside the region of allowed states (see Figure 25.2.1–2). When initial conditions are specified only for p (or for α), Abaqus/Explicit will compute α (or p) assuming that the $p - \alpha$ state lies on the primary (monotonic loading) curve.

Input File Usage: Use some or all of the following options, as required:

*INITIAL CONDITIONS, TYPE=SPECIFIC ENERGY
*INITIAL CONDITIONS, TYPE=STRESS
*INITIAL CONDITIONS, TYPE=POROSITY

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy and initial porosity are not supported in Abaqus/CAE.

JWL high explosive equation of state

The Jones-Wilkens-Lee (or JWL) equation of state models the pressure generated by the release of chemical energy in an explosive. This model is implemented in a form referred to as a programmed burn, which means that the reaction and initiation of the explosive is not determined by shock in the material. Instead, the initiation time is determined by a geometric construction using the detonation wave speed and the distance of the material point from the detonation points.

The JWL equation of state can be written in terms of the internal energy per unit mass, E_m, as

$$p = A\left(1 - \frac{\omega\rho}{R_1\rho_0}\right)\exp\left(-R_1\frac{\rho_0}{\rho}\right) + B\left(1 - \frac{\omega\rho}{R_2\rho_0}\right)\exp\left(-R_2\frac{\rho_0}{\rho}\right) + \omega\rho E_m,$$

where A, B, R_1, R_2, and ω are user-defined material constants; ρ_0 is the user-defined density of the explosive; and ρ is the density of the detonation products.

Input File Usage: Use both of the following options:

*DENSITY *(to specify the density of the explosive ρ_0)*
*EOS, TYPE=JWL *(to specify the material constants A, B, R_1, R_2, and ω)*

Abaqus/CAE Usage: Property module: material editor:
General→Density *(to specify the density of the explosive ρ_0)*
Mechanical→Eos: **Type: JWL** *(to specify the material constants A, B, R_1, R_2, and ω)*

Arrival time of detonation wave

Abaqus/Explicit calculates the arrival time of the detonation wave at a material point (t_d^{mp}) as the distance from the material point to the nearest detonation point divided by the detonation wave speed:

$$t_d^{mp} = \min\left[t_d^N + \sqrt{(\mathbf{x}^{mp} - \mathbf{x}_d^N)\cdot(\mathbf{x}^{mp} - \mathbf{x}_d^N)}\Big/C_d\right],$$

where $\mathbf{x}^{mp}$ is the position of the material point, $\mathbf{x}_d^N$ is the position of the Nth detonation point, t_d^N is the detonation delay time of the Nth detonation point, and C_d is the detonation wave speed of the explosive material. The minimum in the above formula is over the N detonation points that apply to the material point.

Burn fraction

To spread the burn wave over several elements, a burn fraction, F_b, is computed as

$$F_b = \min\left[1, \frac{(t - t_d^{mp})C_d}{B_s l_e}\right],$$

where B_s is a constant that controls the width of the burn wave (set to a value of 2.5) and l_e is the characteristic length of the element. If the time is less than t_d^{mp}, the pressure is zero in the explosive; otherwise, the pressure is given by the product of F_b and the pressure determined from the JWL equation above.

Defining detonation points

You can define any number of detonation points for the explosive material. Coordinates of the points must be defined along with a detonation delay time. Each material point responds to the first detonation point that it sees. The detonation arrival time at a material point is based upon the time that it takes a detonation wave (traveling at the detonation wave speed C_d) to reach the material point plus the detonation delay time for the detonation point. If there are multiple detonation points, the arrival time is based on the minimum arrival time for all the detonation points. In a body with curved surfaces care should be taken that the detonation arrival times are meaningful. The detonation arrival times are based on the straight line of sight from the material point to the detonation point. In a curved body the line of sight may pass outside of the body.

Input File Usage: Use both of the following options to define the detonation points:

∗EOS, TYPE=JWL
∗DETONATION POINT

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Eos**: **Type: JWL**: **Suboptions→Detonation Point**

Initial state

Explosive materials generally have some nominal volumetric stiffness before detonation. It may be useful to incorporate this stiffness when elements modeled with a JWL equation of state are subjected to stress before initiation of detonation by the arriving detonation wave. You can define the pre-detonation bulk modulus, K_{pd}. The pressure will be computed from the volumetric strain and K_{pd} until detonation, at which time the pressure will be determined by the procedure outlined above. The initial relative density (ρ/ρ_0) used in the JWL equation is assumed to be unity. The initial specific energy E_{m_0} is assumed to be equal to the user-defined detonation energy E_0.

If you specify a nonzero value of K_{pd}, you can also define an initial stress state for the explosive materials.

Input File Usage: Use the following option to define the initial stress:

∗INITIAL CONDITIONS, TYPE=STRESS

Optionally, you can also define the initial specific energy directly:

∗INITIAL CONDITIONS, TYPE=SPECIFIC ENERGY

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

Ignition and growth equation of state

The ignition and growth equation of state models shock initiation and detonation wave propagation of solid high explosives. The heterogeneous explosive is modeled as a homogeneous mixture of two phases: the unreacted solid explosive and the reacted gas products. Separate JWL equations of state are prescribed for each phase:

$$p_s = \tilde{F}_{1s}(\rho_s) - \tilde{F}_{1s}(\rho_0) + \tilde{F}_{2s}(\rho_s)E_{ms},$$

$$p_g = \tilde{F}_{1g}(\rho_g) + \tilde{F}_{2g}(\rho_g)\left(E_{mg} + E_d\right),$$

where

$$\tilde{F}_{1i}\left(\rho_i\right) = A_i\left(1 - \frac{\omega_i\rho_i}{R_{1i}\rho_0}\right)\exp\left(-R_{1i}\frac{\rho_0}{\rho_i}\right) + B_i\left(1 - \frac{\omega_i\rho_i}{R_{2i}\rho_0}\right)\exp\left(-R_{2i}\frac{\rho_0}{\rho_i}\right)$$

and

$$\tilde{F}_{2i}\left(\rho_i\right) = \omega_i\rho_i, \quad (i = s, g).$$

The subscript ***s*** refers to the unreacted solid explosive, and ***g*** refers to the reacted gas products. A_i, B_i, R_{1i}, R_{2i}, and ω_i are user-defined material constants used in the JWL equations; E_d is the detonation energy; ρ_0 is the user-defined reference density of the explosive, and ρ_i is the density of the unreacted explosive or the reacted products.

Input File Usage: Use both of the following options:

∗DENSITY*(to specify the density of the explosive ρ_0)*
∗EOS, TYPE=IGNITION AND GROWTH, DETONATION ENERGY=E_d
(to specify the material constants A, B, R_1, R_2, and ω of the unreacted solid explosive and the reacted gas product)

Abaqus/CAE Usage: Property module: material editor:
General→Density *(to specify the density of the explosive ρ_0)*
Mechanical→Eos: **Type: Ignition and growth**: **Detonation energy**: E_d; **Solid Phase** tabbed page and **Gas Phase** tabbed page
(to specify the material constants A, B, R_1, R_2, and ω of the unreacted solid explosive and the reacted gas product)

The mass fraction

The mixture of unreacted solid explosive and reacted gas products is defined by the mass fraction

$$F_i = \frac{m_i}{(m_s + m_g)},$$

where m_s is the mass of the unreacted explosive, and m_g is the mass of the reacted products. It is assumed that the two phases are in thermo-mechanical equilibrium:

$$p_s = p_g \quad \text{and} \quad T_s = T_g.$$

It is also assumed that the volumes are additive:

$$V = V_s + V_g \quad \text{or} \quad \frac{1}{\rho} = (1 - F)\frac{1}{\rho_s} + F\frac{1}{\rho_g}.$$

Similarly, the internal energy is assumed to be additive:

$$(m_s + m_g)\, E_m = m_s E_{ms} + m_g E_{mg},$$

where

$$E_m = E_{m_0} + \int_{\theta_0 - \theta^Z}^{\theta - \theta^Z} c_v(T) dT.$$

Hence, the specific heat of the mixture is given by

$$c_v = (1 - F)c_{vs} + F c_{vg}.$$

Input File Usage: Use the following options to define the specific heat of the unreacted solid explosive:

∗EOS, TYPE=IGNITION AND GROWTH
∗SPECIFIC HEAT, DEPENDENCIES=*n*

Use the following options to define the specific heat of the reacted gas product:

∗EOS, TYPE=IGNITION AND GROWTH
∗GAS SPECIFIC HEAT, DEPENDENCIES=*n*

Abaqus/CAE Usage: Use the following options to define the specific heat of the unreacted solid explosive:

Property module: material editor:
Mechanical→Eos: **Type: Ignition and GrowthThermal→Specific Heat**

Use the following options to define the specific heat of the reacted gas product:

Property module: material editor:

Mechanical→Eos: **Type: Ignition and growth**:
Gas Specific tabbed page: **Specific Heat**

You can toggle on **Use temperature-dependent data** to define the specific heat as a function of temperature and/or select the **Number of field variables** to define the specific heat as a function of field variables.

The reaction rate

The conversion of unreacted solid explosive to reacted gas products is governed by the reaction rate. The reaction rate equation in the ignition and growth model is a pressure-driven rule, which includes three terms:

$$\frac{dF}{dt} = \dot{F}_{ig} + \dot{F}_{G_1} + \dot{F}_{G_2}.$$

These three terms are defined as follows:

$$\begin{aligned}
\dot{F}_{ig} &= I(1-F)^b \left(\frac{\rho_s}{\rho_0} - 1 - a\right)^x, \\
\dot{F}_{G_1} &= G_1(1-F)^c F^d p^y, \\
\dot{F}_{G_2} &= G_2(1-F)^e F^g p^z,
\end{aligned}$$

where $I, G_1, G_2, a, b, c, d, e, g, x, y$, and z are reaction rate constants.

The first term, $\dot{F}_{ig}$, describes hot spot ignition by igniting some of the material relatively quickly but limiting it to a small proportion of the total solid F_{ig}^{max}. The second term, $\dot{F}_{G_1}$, represents the growth of reaction from the hot spot sites into the material and describes the inward and outward grain burning phenomena; this term is limited to a proportion of the total solid $F_{G_1}^{max}$. The third term, $\dot{F}_{G_2}$, is used to describe the rapid transition to detonation observed in some energetic materials.

$$\begin{aligned}
\dot{F}_{ig} &= 0 \text{ if } F \geq F_{ig}^{max} \\
\dot{F}_{G_1} &= 0 \text{ if } F \geq F_{G_1}^{max} \\
\dot{F}_{G_2} &= 0 \text{ if } F \leq F_{G_2}^{min}
\end{aligned}$$

Input File Usage: Use both of the following options to define the reaction rate:

*EOS, TYPE=IGNITION AND GROWTH
*REACTION RATE

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Eos: **Type: Ignition and growth**:
Reaction Rate tabbed page

Initial state

The initial mass fraction of the unreacted solid explosive is assumed to be one. The initial relative density (ρ/ρ_0) used in the ignition and growth equation is assumed to be unity. The initial specific energy and initial stress can be defined for the unreacted explosive.

Input File Usage: Use the following option to define the initial stress:

*INITIAL CONDITIONS, TYPE=STRESS

Optionally, you can also define the initial specific energy directly:

*INITIAL CONDITIONS, TYPE=SPECIFIC ENERGY

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

Ideal gas equation of state

An ideal gas equation of state can be written in the form of

$$p + p_A = \rho R \left(\theta - \theta^Z\right),$$

where p_A is the ambient pressure, $\boldsymbol{R}$ is the gas constant, θ is the current temperature, and θ^Z is the absolute zero on the temperature scale being used. It is an idealization to real gas behavior and can be used to model any gases approximately under appropriate conditions (e.g., low pressure and high temperature).

One of the important features of an ideal gas is that its specific energy depends only upon its temperature; therefore, the specific energy can be integrated numerically as

$$E_m = E_{m_0} + \int_{\theta_0-\theta^Z}^{\theta-\theta^Z} c_v(T)dT,$$

where E_{m_0} is the initial specific energy at the initial temperature θ_0 and c_v is the specific heat at constant volume (or the constant volume heat capacity), which depends only upon temperature for an ideal gas.

Modeling with an ideal gas equation of state is typically performed adiabatically; the temperature increase is calculated directly at the material integration points according to the adiabatic thermal energy increase caused by the work pdv, where $\boldsymbol{v}$ is the specific volume (the volume per unit mass, $v = 1/\rho$). Therefore, unless a fully coupled temperature-displacement analysis is performed, an adiabatic condition is always assumed in Abaqus/Explicit.

When performing a fully coupled temperature-displacement analysis, the pressure stress and specific energy are updated based on the evolving temperature field. The energy increase due to the change in state will be accounted for in the heat equation and will be subject to heat conduction.

For the ideal gas model in Abaqus/Explicit you define the gas constant, $\boldsymbol{R}$, and the ambient pressure, p_A. For an ideal gas $\boldsymbol{R}$ can be determined from the universal gas constant, $\tilde{R}$, and the molecular weight, MW, as follows:

$$R = \frac{\tilde{R}}{MW}.$$

In general, the value R for any gas can be estimated by plotting $pv/(\theta - \theta^Z)$ as a function of state (e.g., pressure or temperature). The ideal gas approximation is adequate in any region where this value is constant. You must specify the specific heat at constant volume, c_v. For an ideal gas c_v is related to the specific heat at constant pressure, c_p, by

$$R = c_p - c_v.$$

Input File Usage: Use both of the following options:

*EOS, TYPE=IDEAL GAS
*SPECIFIC HEAT, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Eos: **Type: Ideal Gas**
Thermal→Specific Heat

Initial state

There are different methods to define the initial state of the gas. You can specify the initial density, ρ, and either the initial pressure stress, p_0, or the initial temperature, θ_0. The initial value of the unspecified field (temperature or pressure) is determined from the equation of state. Alternatively, you can specify both the initial pressure stress and the initial temperature. In this case the user-specified initial density is replaced by that derived from the equation of state in terms of initial pressure and temperature.

By default, Abaqus/Explicit automatically computes the initial specific energy, E_{m_0}, from the initial temperature by numerically integrating the equation

$$E_{m_0} = \int_0^{\theta_0 - \theta^Z} c_v(T) dT.$$

Optionally, you can override this default behavior by defining the initial specific energy for the ideal gas directly.

Input File Usage: Use some or all of the following options, as required:

*DENSITY, DEPENDENCIES=*n*
*INITIAL CONDITIONS, TYPE=STRESS
*INITIAL CONDITIONS, TYPE=TEMPERATURE

Use the following option to specify the initial specific energy directly:

*INITIAL CONDITIONS, TYPE=SPECIFIC ENERGY

Abaqus/CAE Usage: Property module: material editor: **General→Density**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Temperature** for the **Types for Selected Step**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

The value of absolute zero

When a non-absolute temperature scale is used, you must specify the value of absolute zero temperature.

Input File Usage: *PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Any module: **Model→Edit Attributes→*model_name***: **Absolute zero temperature**

A special case

In the case of an adiabatic analysis with constant specific heat (both c_v and c_p are constant), the specific energy is linear in temperature

$$E_m = c_v \left(\theta - \theta^Z\right).$$

The pressure stress can, therefore, be recast in the common form of

$$p + p_A = (\gamma - 1)\rho E_m,$$

where $\gamma = c_p/c_v$ is the ratio of specific heats and can be defined as

$$\gamma = \frac{n+2}{n},$$

where

$n = 3$	for a monatomic;
$n = 5$	for a diatomic; and
$n = 6$	for a polyatomic gas.

Comparison with the hydrostatic fluid model

The ideal gas equation of state can be used to model wave propagation effects and the dynamics of a spatially varying state of a gaseous region. For cases in which the inertial effects of the gas are not important and the state of the gas can be assumed to be uniform throughout a region, the hydrostatic fluid model ("Surface-based fluid cavities: overview," Section 11.5.1) is a simpler, more computationally efficient alternative.

Deviatoric behavior

The equation of state defines only the material's hydrostatic behavior. It can be used by itself, in which case the material has only volumetric strength (the material is assumed to have no shear strength). Alternatively, Abaqus/Explicit allows you to define deviatoric behavior, assuming that the deviatoric and volumetric responses are uncoupled. Two models are available for the deviatoric response: a linear isotropic elastic model and a viscous model. The material's volumetric response is governed then by the equation of state model, while its deviatoric response is governed by either the linear isotropic elastic model or the viscous fluid model.

Elastic shear behavior

For the elastic shear behavior the deviatoric stress is related to the deviatoric strain as

$$\mathbf{S} = 2\mu\mathbf{e}^{el},$$

where $\mathbf{S}$ is the deviatoric stress and $\mathbf{e}^{el}$ is the deviatoric elastic strain. See "Defining isotropic shear elasticity for equations of state in Abaqus/Explicit" in "Linear elastic behavior," Section 22.2.1, for more details.

Input File Usage: Use both of the following options to define elastic shear behavior:

*EOS
*ELASTIC, TYPE=SHEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Elasticity→Elastic**; **Type: Shear**; **Shear Modulus**

Viscous shear behavior

For the viscous shear behavior the deviatoric stress is related to the deviatoric strain rate as

$$\mathbf{S} = 2\eta\dot{\mathbf{e}} = \eta\dot{\gamma},$$

where $\mathbf{S}$ is the deviatoric stress, $\dot{\mathbf{e}}$ is the deviatoric part of the strain rate, η is the viscosity, and $\dot{\gamma} = 2\dot{\mathbf{e}}$ is the engineering shear strain rate.

Abaqus/Explicit provides a wide range of viscosity models to describe both Newtonian and non-Newtonian fluids. These are described in "Viscosity," Section 26.1.4.

Input File Usage: Use both of the following options to define viscous shear behavior:

*EOS
*VISCOSITY

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Viscosity**

Use with the Mises or the Johnson-Cook plasticity models

An equation of state model can be used with the Mises ("Classical metal plasticity," Section 23.2.1) or the Johnson-Cook ("Johnson-Cook plasticity," Section 23.2.7) plasticity models to model elastic-plastic behavior. In this case you must define the elastic part of the shear behavior. The material's volumetric response is governed by the equation of state model, while the deviatoric response is governed by the linear elastic shear and the plasticity model.

Input File Usage: Use the following options:

*EOS
*ELASTIC, TYPE=SHEAR
*PLASTIC

Abaqus/CAE Usage: Property module: material editor:
Mechanical→Elasticity→Elastic; **Type: Shear**
Mechanical→Plasticity→Plastic

Initial conditions

You can specify initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use with the extended Drucker-Prager plasticity models

An equation of state model can be used in conjunction with the extended Drucker-Prager ("Extended Drucker-Prager models," Section 23.3.1) plasticity models to model pressure-dependent plasticity behavior. This approach can be appropriate for modeling the response of ceramics and other brittle materials under high velocity impact conditions. In this case you must define the elastic part of the shear behavior. The material's deviatoric response is governed by the linear elastic shear and the pressure-dependent plasticity model, while the volumetric response is governed by the equation of state model. In particular, no plastic dilation effects are taken into account (if you specify a dilation angle other than zero, the value is ignored and Abaqus/Explicit issues a warning message).

"High-velocity impact of a ceramic target," Section 2.1.18 of the Abaqus Example Problems Manual illustrates the use of an equation of state model with the extended Drucker-Prager plasticity models.

Input File Usage: Use the following options:

*EOS
*ELASTIC, TYPE=SHEAR
*DRUCKER PRAGER
*DRUCKER PRAGER HARDENING

Abaqus/CAE Usage: Property module: material editor:

Mechanical→Elasticity→Elastic; **Type: Shear**
Mechanical→Plasticity→Drucker Prager: **Suboptions→Drucker Prager Hardening**

Initial conditions

You can specify initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1).

Input File Usage: *INITIAL CONDITIONS, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use with the tensile failure model

An equation of state model (except the ideal gas equation of state) can also be used with the tensile failure model ("Dynamic failure models," Section 23.2.8) to model dynamic spall or a pressure cutoff. The tensile failure model uses the hydrostatic pressure stress as a failure measure and offers a number of failure choices. You must provide the hydrostatic cutoff stress.

You can specify that the deviatoric stresses should fail when the tensile failure criterion is met. In the case where the material's deviatoric behavior is not defined, this specification is meaningless and is, therefore, ignored.

The tensile failure model in Abaqus/Explicit is designed for high-strain-rate dynamic problems in which inertia effects are important. Therefore, it should be used only for such situations. Improper use of the tensile failure model may result in an incorrect simulation.

Input File Usage: Use the following options:

*EOS
*TENSILE FAILURE

Abaqus/CAE Usage: The tensile failure model is not supported in Abaqus/CAE.

Adiabatic assumption

An adiabatic condition is always assumed for materials modeled with an equation of state unless a dynamic coupled temperature-displacement procedure is used. The adiabatic condition is assumed irrespective of whether an adiabatic dynamic stress analysis step has been specified. The temperature increase is calculated directly at the material integration points according to the adiabatic thermal energy increase caused by the mechanical work

$$\rho c_v(\theta)\frac{\partial \theta}{\partial t} = (p - p_{bv})\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \mathbf{S} : \dot{\mathbf{e}},$$

where c_v is the specific heat at constant volume. Specifying temperature as a predefined field has no effect on the behavior of this model.

When performing a fully coupled temperature-displacement analysis, the specific energy is updated based on the evolving temperature field using

$$\rho \frac{\partial E_m}{\partial t} = \rho c_v(\theta) \frac{\partial \theta}{\partial t}.$$

Modeling fluids

A linear $U_s - U_p$ equation of state model can be used to model incompressible viscous and inviscid laminar flow governed by the Navier-Stokes equation of motion. The volumetric response is governed by the equations of state, where the bulk modulus acts as a penalty parameter for the incompressible constraint.

To model a viscous laminar flow that follows the Navier-Poisson law of a Newtonian fluid, use the Newtonian viscous deviatoric model and define the viscosity as the real linear viscosity of the fluid. To model non-Newtonian viscous flow, use one of the nonlinear viscosity models available in Abaqus/Explicit. Appropriate initial conditions for velocity and stress are essential to get an accurate solution for this class of problems.

To model an incompressible inviscid fluid such as water in Abaqus/Explicit, it is useful to define a small amount of shear resistance to suppress shear modes that can otherwise tangle the mesh. Here the shear stiffness or shear viscosity acts as a penalty parameter. The shear modulus or viscosity should be small because flow is inviscid; a high shear modulus or viscosity will result in an overly stiff response. To avoid an overly stiff response, the internal forces arising due to the deviatoric response of the material should be kept several orders of magnitude below the forces arising due to the volumetric response. This can be done by choosing an elastic shear modulus that is several orders of magnitude lower than the bulk modulus. If the viscous model is used, the shear viscosity specified should be on the order of the shear modulus, calculated as above, scaled by the stable time increment. The expected stable time increment can be obtained from a data check analysis of the model. This method is a convenient way to approximate a shear resistance that will not introduce excessive viscosity in the material.

If a shear model is defined, the hourglass control forces are calculated based on the shear resistance of the material. Thus, in materials with extremely low or zero shear strengths such as inviscid fluids, the hourglass forces calculated based on the default parameters are insufficient to prevent spurious hourglass modes. Therefore, a sufficiently high hourglass scaling factor is recommended to increase the resistance to such modes.

Elements

Equations of state can be used with any solid (continuum) elements in Abaqus/Explicit except plane stress elements. For three-dimensional applications exhibiting high confinement, the default kinematic formulation is recommended with reduced-integration solid elements (see “Section controls,” Section 27.1.4).

Output

In addition to the standard output identifiers available in Abaqus (“Abaqus/Explicit output variable identifiers,” Section 4.2.2), the following variables have special meaning for the equation of state models:

PALPH	Distension, α, of the $P-\alpha$ porous material. The current porosity is equal to one minus the inverse of α: $n = 1 - \alpha^{-1}$.
PALPHMIN	Minimum value, α_{min}, of the distension attained during plastic compaction of the $P-\alpha$ porous material.
PEEQ	Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}|_0 + \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\bar{\varepsilon}^{pl}|_0$ is the initial equivalent plastic strain (zero or user-specified; see "Initial conditions"). This is relevant only if the equation of state model is used in combination with the Mises, Johnson-Cook, or extended Drucker-Prage plasticity models.

Additional references

- Carroll, M., and A. C. Holt, "Suggested Modification of the $P-\alpha$ Model for Porous Materials," Journal of Applied Physics, vol. 43, no. 2, pp. 759–761, 1972.
- Dobratz, B. M., "LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants," UCRL-52997, Lawrence Livermore National Laboratory, Livermore, California, January 1981.
- Herrmann, W., "Constitutive Equation for the Dynamic Compaction of Ductile Porous Materials," Journal of Applied Physics, vol. 40, no. 6, pp. 2490–2499, 1968.
- Lee, E., M. Finger, and W. Collins, "JWL Equation of State Coefficients for High Explosives," UCID-16189, Lawrence Livermore National Laboratory, Livermore, California, January 1973.
- Wardlaw, A. B., R. McKeown, and H. Chen, "Implementation and Application of the $P-\alpha$ Equation of State in the DYSMAS Code," Naval Surface Warfare Center, Dahlgren Division, Report Number: NSWCDD/TR-95/107, May 1996.

26. Other Material Properties

Mechanical properties 26.1
Heat transfer properties 26.2
Acoustic properties 26.3
Mass diffusion properties 26.4
Electromagnetic properties 26.5
Pore fluid flow properties 26.6
User materials 26.7

26.1 Mechanical properties

- “Material damping,” Section 26.1.1
- “Thermal expansion,” Section 26.1.2
- “Field expansion,” Section 26.1.3
- “Viscosity,” Section 26.1.4

26.1.1 MATERIAL DAMPING

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Dynamic analysis procedures: overview," Section 6.3.1
- "Material library: overview," Section 21.1.1
- *DAMPING
- "Defining damping" in "Defining other mechanical models," Section 12.9.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Material damping can be defined:

- for direct-integration (nonlinear, implicit or explicit), subspace-based direct-integration, direct-solution steady-state, and subspace-based steady-state dynamic analysis; or
- for mode-based (linear) dynamic analysis in Abaqus/Standard.

Rayleigh damping

In direct-integration dynamic analysis you very often define energy dissipation mechanisms—dashpots, inelastic material behavior, etc.—as part of the basic model. In such cases there is usually no need to introduce additional damping: it is often unimportant compared to these other dissipative effects. However, some models do not have such dissipation sources (an example is a linear system with chattering contact, such as a pipeline in a seismic event). In such cases it is often desirable to introduce some general damping. Abaqus provides "Rayleigh" damping for this purpose. It provides a convenient abstraction to damp lower (mass-dependent) and higher (stiffness-dependent) frequency range behavior.

Rayleigh damping can also be used in direct-solution steady-state dynamic analyses and subspace-based steady-state dynamic analyses to get quantitatively accurate results, especially near natural frequencies.

To define material Rayleigh damping, you specify two Rayleigh damping factors: α_R for mass proportional damping and β_R for stiffness proportional damping. In general, damping is a material property specified as part of the material definition. For the cases of rotary inertia, point mass elements, and substructures, where there is no reference to a material definition, the damping can be defined in conjunction with the property references. Any mass proportional damping also applies to nonstructural features (see "Nonstructural mass definition," Section 2.7.1).

For a given mode i the fraction of critical damping, ξ_i, can be expressed in terms of the damping factors α_R and β_R as:

$$\xi_i = \frac{\alpha_R}{2\omega_i} + \frac{\beta_R \omega_i}{2},$$

where ω_i is the natural frequency at this mode. This equation implies that, generally speaking, the mass proportional Rayleigh damping, α_R, damps the lower frequencies and the stiffness proportional Rayleigh damping, β_R, damps the higher frequencies.

Mass proportional damping

The α_R factor introduces damping forces caused by the absolute velocities of the model and so simulates the idea of the model moving through a viscous "ether" (a permeating, still fluid, so that any motion of any point in the model causes damping). This damping factor defines mass proportional damping, in the sense that it gives a damping contribution proportional to the mass matrix for an element. If the element contains more than one material in Abaqus/Standard, the volume average value of α_R is used to multiply the element's mass matrix to define the damping contribution from this term. If the element contains more than one material in Abaqus/Explicit, the mass average value of α_R is used to multiply the element's lumped mass matrix to define the damping contribution from this term. α_R has units of (1/time).

Input File Usage: *DAMPING, ALPHA=α_R

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damping**: **Alpha**: α_R

Defining variable mass proportional damping in Abaqus/Explicit

In Abaqus/Explicit you can define α_R as a tabular function of temperature and/or field variables. Therefore, mass proportional damping can vary during an Abaqus/Explicit analysis.

Input File Usage: *DAMPING, ALPHA=TABULAR

Stiffness proportional damping

The β_R factor introduces damping proportional to the strain rate, which can be thought of as damping associated with the material itself. β_R defines damping proportional to the elastic material stiffness. Since the model may have quite general nonlinear response, the concept of "stiffness proportional damping" must be generalized, since it is possible for the tangent stiffness matrix to have negative eigenvalues (which would imply negative damping). To overcome this problem, β_R is interpreted as defining viscous material damping in Abaqus, which creates an additional "damping stress," $\boldsymbol{\sigma}_d$, proportional to the total strain rate:

$$\boldsymbol{\sigma}_d = \beta_R \mathbf{D}^{el} \dot{\boldsymbol{\varepsilon}},$$

where $\dot{\boldsymbol{\varepsilon}}$ is the strain rate. For hyperelastic ("Hyperelastic behavior of rubberlike materials," Section 22.5.1) and hyperfoam ("Hyperelastic behavior in elastomeric foams," Section 22.5.2) materials $\mathbf{D}^{el}$ is defined as the elastic stiffness in the strain-free state. For all other linear elastic materials in Abaqus/Standard and all other materials in Abaqus/Explicit, $\mathbf{D}^{el}$ is the material's current elastic stiffness. $\mathbf{D}^{el}$ will be calculated based on the current temperature during the analysis.

This damping stress is added to the stress caused by the constitutive response at the integration point when the dynamic equilibrium equations are formed, but it is not included in the stress output. As a result, damping can be introduced for any nonlinear case and provides standard Rayleigh damping for linear

cases; for a linear case stiffness proportional damping is exactly the same as defining a damping matrix equal to β_R times the (elastic) material stiffness matrix. Other contributions to the stiffness matrix (e.g., hourglass, transverse shear, and drill stiffnesses) are not included when computing stiffness proportional damping. β_R has units of (time).

Input File Usage: *DAMPING, BETA=β_R

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damping**: **Beta**: β_R

Defining variable stiffness proportional damping in Abaqus/Explicit

In Abaqus/Explicit you can define β_R as a tabular function of temperature and/or field variables. Therefore, stiffness proportional damping can vary during an Abaqus/Explicit analysis.

Input File Usage: *DAMPING, BETA=TABULAR

Structural damping

Structural damping assumes that the damping forces are proportional to the forces caused by stressing of the structure and are opposed to the velocity. Therefore, this form of damping can be used only when the displacement and velocity are exactly 90° out of phase. Structural damping is best suited for frequency domain dynamic procedures (see "Damping in modal superposition procedures" below). The damping forces are then

$$F_D^N = isI^N,$$

where F_D^N are the damping forces, $i = \sqrt{-1}$, s is the user-defined structural damping factor, and I^N are the forces caused by stressing of the structure. The damping forces due to structural damping are intended to represent frictional effects (as distinct from viscous effects). Thus, structural damping is suggested for models involving materials that exhibit frictional behavior or where local frictional effects are present throughout the model, such as dry rubbing of joints in a multi-link structure.

Structural damping can be added to the model as mechanical dampers such as connector damping or as a complex stiffness on spring elements.

Structural damping can be used in steady-state dynamic procedures that allow for nondiagonal damping.

Input File Usage: Use the following option to define structural damping:

*DAMPING, STRUCTURAL=s

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Damping**: **Structural**: s

Artificial damping in direct-integration dynamic analysis

In Abaqus/Standard the operators used for implicit direct time integration introduce some artificial damping in addition to Rayleigh damping. Damping associated with the Hilber-Hughes-Taylor and hybrid operators is usually controlled by the Hilber-Hughes-Taylor parameter α, which is not the same as the α_R parameter controlling the mass proportional part of Rayleigh damping. The β and γ parameters of the Hilber-Hughes-Taylor and hybrid operators also affect numerical damping. The α,

β, and γ parameters are not available for the backward Euler operator. See "Implicit dynamic analysis using direct integration," Section 6.3.2, for more information about this other form of damping.

Artificial damping in explicit dynamic analysis

Rayleigh damping is meant to reflect physical damping in the actual material. In Abaqus/Explicit a small amount of numerical damping is introduced by default in the form of bulk viscosity to control high frequency oscillations; see "Explicit dynamic analysis," Section 6.3.3, for more information about this other form of damping.

Effects of damping on the stable time increment in Abaqus/Explicit

As the fraction of critical damping for the highest mode ($\xi_{\max}$) increases, the stable time increment for Abaqus/Explicit decreases according to the equation

$$\Delta t \leq \frac{2}{\omega_{\max}}(\sqrt{1+\xi_{\max}^2}-\xi_{\max}),$$

where (by substituting $\omega_{\max}$, the frequency of the highest mode, into the equation for ξ_i given previously)

$$\xi_{\max} = \frac{\alpha_R}{2\omega_{\max}} + \frac{\beta_R \omega_{\max}}{2}.$$

These equations indicate a tendency for stiffness proportional damping to have a greater effect on the stable time increment than mass proportional damping.

To illustrate the effect that damping has on the stable time increment, consider a cantilever in bending modeled with continuum elements. The lowest frequency is $\omega_{\min} = 1$ rad/sec, while for the particular mesh chosen, the highest frequency is $\omega_{\max} = 1000$ rad/sec. The lowest mode in this problem corresponds to the cantilever in bending, and the highest frequency is related to the dilation of a single element.

With no damping the stable time increment is

$$\Delta t = \frac{2}{\omega_{\max}} = 2 \times 10^{-3}\text{sec}.$$

If we use stiffness proportional damping to create 1% of critical damping in the lowest mode, the damping factor is given by

$$\beta_R = \frac{2 \times 0.01}{1} = 2 \times 10^{-2}\text{sec}.$$

This corresponds to a critical damping factor in the highest mode of

$$\xi_{\max} = \frac{\omega_{\max}\beta_R}{2} = 10.$$

The stable time increment with damping is, thus, reduced by a factor of

$$(\sqrt{1+10^2}-10) \approx 0.05,$$

and becomes

$$\begin{aligned}\Delta t &\approx (2\times 10^{-3})\times 0.05\\ &\approx 1\times 10^{-4}.\end{aligned}$$

Thus, introducing 1% critical damping in the lowest mode reduces the stable time increment by a factor of twenty.

However, if we use mass proportional damping to damp out the lowest mode with 1% of critical damping, the damping factor is given by

$$\alpha_R = 2\omega_{\min}\xi = 2\times 1\times 10^{-2} = 2\times 10^{-2}\text{sec}^{-1},$$

which corresponds to a critical damping factor in the highest mode of

$$\xi_{\max} = \frac{\alpha_R}{2\omega_{\max}} = \frac{2\times 10^{-2}}{2\times 1000} = 10^{-5}.$$

The stable time increment with damping is reduced by a factor of

$$(\sqrt{1+10^{-10}}-10^{-5}) \approx 0.99999,$$

which is almost negligible.

This example demonstrates that it is generally preferable to damp out low frequency response with mass proportional damping rather than stiffness proportional damping. However, mass proportional damping can significantly affect rigid body motion, so large α_R is often undesirable. To avoid a dramatic drop in the stable time increment, the stiffness proportional damping factor, β_R, should be less than or of the same order of magnitude as the initial stable time increment without damping. With $\beta_R = 2/\omega_{\max}$, the stable time increment is reduced by about 52%.

Damping in modal superposition procedures

Damping can be specified as part of the step definition for modal superposition procedures. “Damping in a linear dynamic analysis” in “Dynamic analysis procedures: overview,” Section 6.3.1, describes the availability of damping types, which depends on the procedure type and the architecture used to perform the analysis, and provides details on the following types of damping:

- Viscous modal damping (Rayleigh damping and fraction of critical damping)
- Structural modal damping
- Composite modal damping

Use with other material models

The β_R factor applies to all elements that use a linear elastic material definition ("Linear elastic behavior," Section 22.2.1) and to Abaqus/Standard beam and shell elements that use general sections. In the latter case, if a nonlinear beam section definition is provided, the β_R factor is multiplied by the slope of the force-strain (or moment-curvature) relationship at zero strain or curvature. In addition, the β_R factor applies to all Abaqus/Explicit elements that use a hyperelastic material definition ("Hyperelastic behavior of rubberlike materials," Section 22.5.1), a hyperfoam material definition ("Hyperelastic behavior in elastomeric foams," Section 22.5.2), or general shell sections ("Using a general shell section to define the section behavior," Section 29.6.6).

In the case of a no tension elastic material the β_R factor is not used in tension, while for a no compression elastic material the β_R factor is not used in compression (see "No compression or no tension," Section 22.2.2). In other words, these modified elasticity models exhibit damping only when they have stiffness.

Elements

The α_R factor is applied to all elements that have mass including point mass elements (discrete DASHPOTA elements in each global direction, each with one node fixed, can also be used to introduce this type of damping). For point mass and rotary inertia elements mass proportional or composite modal damping are defined as part of the point mass or rotary inertia definitions ("Point masses," Section 30.1.1, and "Rotary inertia," Section 30.2.1).

The β_R factor is not available for spring elements: discrete dashpot elements should be used in parallel with spring elements instead.

The β_R factor is also not applied to the transverse shear terms in Abaqus/Standard beams and shells.

In Abaqus/Standard composite modal damping cannot be used with or within substructures. Rayleigh damping can be introduced for substructures. When Rayleigh damping is used within a substructure, α_R and β_R are averaged over the substructure to define single values of α_R and β_R for the substructure. These are weighted averages, using the mass as the weighting factor for α_R and the volume as the weighting factor for β_R. These averaged damping values can be superseded by providing them directly in a second damping definition. See "Using substructures," Section 10.1.1.

26.1.2 THERMAL EXPANSION

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "UEXPAN," Section 1.1.29 of the Abaqus User Subroutines Reference Manual
- *EXPANSION
- "Defining other mechanical models," Section 12.9.4 of the Abaqus/CAE User's Manual
- "Defining a fluid-filled porous material," Section 12.12.3 of the Abaqus/CAE User's Manual

Overview

Thermal expansion effects:

- can be defined by specifying thermal expansion coefficients so that Abaqus can compute thermal strains and, in Abaqus/CFD, buoyancy forces;
- can be isotropic, orthotropic, or fully anisotropic;
- are defined as total expansion from a reference temperature;
- can be specified as a function of temperature and/or field variables;
- can be defined with a distribution for solid continuum elements in Abaqus/Standard; and
- in Abaqus/Standard can be specified directly in user subroutine **UEXPAN** (if the thermal strains are complicated functions of field variables and state variables).

Defining thermal expansion coefficients

Thermal expansion is a material property included in a material definition (see "Material data definition," Section 21.1.2) except when it refers to the expansion of a gasket whose material properties are not defined as part of a material definition. In that case expansion must be used in conjunction with the gasket behavior definition (see "Defining the gasket behavior directly using a gasket behavior model," Section 32.6.6).

In an Abaqus/Standard analysis a spatially varying thermal expansion can be defined for homogeneous solid continuum elements by using a distribution ("Distribution definition," Section 2.8.1). The distribution must include default values for the thermal expansion. If a distribution is used, no dependencies on temperature and/or field variables for the thermal expansion can be defined.

Input File Usage: Use the following options to define thermal expansion for most materials:

*MATERIAL
*EXPANSION

Use the following options to define thermal expansion for gaskets whose constitutive response is defined directly as gasket behavior:

*GASKET BEHAVIOR
*EXPANSION

Abaqus/CAE Usage: Use the following option in conjunction with other material behaviors, including gasket behavior, to include thermal expansion effects:

Property module: material editor: **Mechanical→Expansion**

Computation of thermal strains

Abaqus requires thermal expansion coefficients, α, that define the total thermal expansion from a reference temperature, θ^0, as shown in Figure 26.1.2–1.

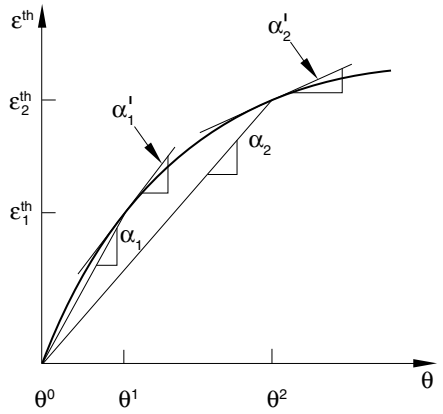


Figure 26.1.2–1 Definition of the thermal expansion coefficient.

They generate thermal strains according to the formula

$$\varepsilon^{th} = \alpha(\theta, f_\beta)(\theta - \theta^0) - \alpha(\theta^I, f_\beta^I)(\theta^I - \theta^0),$$

where

$\alpha(\theta, f_\beta)$	is the thermal expansion coefficient;
θ	is the current temperature;
θ^I	is the initial temperature;
f_β	are the current values of the predefined field variables;
f_β^I	are the initial values of the field variables; and
θ^0	is the reference temperature for the thermal expansion coefficient.

The second term in the above equation represents the strain due to the difference between the initial temperature, θ^I, and the reference temperature, θ^0. This term is necessary to enforce the assumption that

there is no initial thermal strain for cases in which the reference temperature does not equal the initial temperature.

Defining the reference temperature

If the coefficient of thermal expansion, α, is not a function of temperature or field variables, the value of the reference temperature, θ^0, is not needed. If α is a function of temperature or field variables, you can define θ^0.

Input File Usage: *EXPANSION, ZERO=θ^0

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Expansion**: **Reference temperature:** θ^0

Converting thermal expansion coefficients from differential form to total form

Total thermal expansion coefficients are commonly available in tables of material properties. However, sometimes you are given thermal expansion data in differential form:

$$d\varepsilon^{th} = \alpha'(\theta)\, d\theta;$$

that is, the tangent to the strain-temperature curve is provided (see Figure 26.1.2–1). To convert to the total thermal expansion form required by Abaqus, this relationship must be integrated from a suitably chosen reference temperature, θ^0:

$$\varepsilon^{th} = \int_{\theta^0}^{\theta} \alpha'\, d\theta \Rightarrow \alpha(\theta) = \frac{1}{\theta - \theta^0} \int_{\theta^0}^{\theta} \alpha'\, d\theta.$$

For example, suppose α' is a series of constant values: α'_1 between θ^0 and θ^1; α'_2 between θ^1 and θ^2; α'_3 between θ^2 and θ^3; etc. Then,

$$\begin{aligned}
\varepsilon_1^{th} &= \alpha'_1(\theta^1 - \theta^0) \\
\varepsilon_2^{th} &= \varepsilon_1^{th} + \alpha'_2(\theta^2 - \theta^1) \\
\varepsilon_3^{th} &= \varepsilon_2^{th} + \alpha'_3(\theta^3 - \theta^2).
\end{aligned}$$

The corresponding total expansion coefficients required by Abaqus are then obtained as

$$\begin{aligned}
\alpha_1 &= \varepsilon_1^{th}/(\theta^1 - \theta^0) \\
\alpha_2 &= \varepsilon_2^{th}/(\theta^2 - \theta^0) \\
\alpha_3 &= \varepsilon_3^{th}/(\theta^3 - \theta^0).
\end{aligned}$$

Defining increments of thermal strain in user subroutine UEXPAN

Increments of thermal strain can be specified in Abaqus/Standard user subroutine **UEXPAN** as functions of temperature and/or predefined field variables. User subroutine **UEXPAN** must be used if the thermal strain increments depend on state variables.

Input File Usage: *EXPANSION, USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Expansion**: **Use user subroutine UEXPAN**

Defining the initial temperature and field variable values

If the coefficient of thermal expansion, α, is a function of temperature or field variables, the initial temperature and initial field variable values, θ^I and f_β^I, are given as described in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1.

Element removal and reactivation

If an element has been removed and subsequently reactivated in Abaqus/Standard ("Element and contact pair removal and reactivation," Section 11.2.1), θ^I and f_β^I in the equation for the thermal strains represent temperature and field variable values as they were at the moment of reactivation.

Defining directionally dependent thermal expansion

Isotropic or orthotropic thermal expansion can be defined in Abaqus. In addition, fully anisotropic thermal expansion can be defined in Abaqus/Standard.

Orthotropic and anisotropic thermal expansion can be used only with materials where the material directions are defined with local orientations (see "Orientations," Section 2.2.5).

Orthotropic thermal expansion in Abaqus/Explicit is allowed only with anisotropic elasticity (including orthotropic elasticity) and anisotropic yield (see "Anisotropic yield/creep," Section 23.2.6).

Only isotropic thermal expansion is allowed in Abaqus/CFD, for adiabatic stress analysis, and with the hyperelastic and hyperfoam material models.

Isotropic expansion

If the thermal expansion coefficient is defined directly, only one value of α is needed at each temperature. If user subroutine **UEXPAN** is used, only one isotropic thermal strain increment ($\Delta\varepsilon = \Delta\varepsilon_{11} = \Delta\varepsilon_{22} = \Delta\varepsilon_{33}$) must be defined.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

*EXPANSION, TYPE=ISO

Use the following option to define the thermal expansion with user subroutine **UEXPAN**:

*EXPANSION, TYPE=ISO, USER

Abaqus/CAE Usage: Use the following input to define the thermal expansion coefficient directly:

Property module: material editor: **Mechanical→Expansion**: **Type: Isotropic**

Use the following input to define the thermal expansion with user subroutine **UEXPAN**:

Property module: material editor: **Mechanical→Expansion**: **Type: Isotropic**, **Use user subroutine UEXPAN**

Orthotropic expansion

If the thermal expansion coefficients are defined directly, the three expansion coefficients in the principal material directions (α_{11}, α_{22}, and α_{33}) should be given as functions of temperature. If user subroutine **UEXPAN** is used, the three components of thermal strain increment in the principal material directions ($\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, and $\Delta\varepsilon_{33}$) must be defined.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

*EXPANSION, TYPE=ORTHO

Use the following option to define the thermal expansion with user subroutine **UEXPAN**:

*EXPANSION, TYPE=ORTHO, USER

Abaqus/CAE Usage: Use the following input to define the thermal expansion coefficient directly:

Property module: material editor: **Mechanical→Expansion**:
Type: Orthotropic

Use the following input to define the thermal expansion with user subroutine **UEXPAN**:

Property module: material editor: **Mechanical→Expansion**: **Type: Orthotropic**, **Use user subroutine UEXPAN**

Anisotropic expansion

If the thermal expansion coefficients are defined directly, all six components of α (α_{11}, α_{22}, α_{33}, α_{12}, α_{13}, α_{23}) must be given as functions of temperature. If user subroutine **UEXPAN** is used, all six components of the thermal strain increment ($\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, $\Delta\varepsilon_{33}$, $\Delta\varepsilon_{12}$, $\Delta\varepsilon_{13}$, $\Delta\varepsilon_{23}$) must be defined.

In an Abaqus/Standard analysis if a distribution is used to define the thermal expansion, the number of expansion coefficients given for each element in the distribution, which is determined by the associated distribution table ("Distribution definition," Section 2.8.1), must be consistent with the level of anisotropy specified for the expansion behavior. For example, if orthotropic behavior is specified, three expansion coefficients must be defined for each element in the distribution.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

*EXPANSION, TYPE=ANISO

Use the following option to define the thermal expansion with user subroutine **UEXPAN**:

*EXPANSION, TYPE=ANISO, USER

Abaqus/CAE Usage: Use the following input to define the thermal expansion coefficient directly:

Property module: material editor: **Mechanical→Expansion**:
Type: Anisotropic

Use the following input to define the thermal expansion with user subroutine **UEXPAN**:

Property module: material editor: **Mechanical→Expansion**: **Type: Anisotropic**, **Use user subroutine UEXPAN**

Thermal stress

When a structure is not free to expand, a change in temperature will cause stress. For example, consider a single two-node truss of length L that is completely restrained at both ends. The cross-sectional area; the Young's modulus, E; and the thermal expansion coefficient, α, are all constant. The stress in this one-dimensional problem can then be calculated from Hooke's Law as $\sigma_x = E(\varepsilon_x - \varepsilon_x^{th})$, where ε_x is the total strain and $\varepsilon_x^{th} = \alpha\Delta\theta$ is the thermal strain, where $\Delta\theta$ is the temperature change. Since the element is fully restrained, $\varepsilon_x = 0$. If the temperature at both nodes is the same, we obtain the stress $\sigma_x = -E\alpha\Delta\theta$.

Constrained thermal expansion can cause significant stress. For typical structural metals, temperature changes of about 150°C (300°F) can cause yield. Therefore, it is often important to define boundary conditions with particular care for problems involving thermal loading to avoid overconstraining the thermal expansion.

Energy balance considerations

Abaqus does not account for thermal expansion effects in the total energy balance equation, which can lead to an apparent imbalance of the total energy of the model. For example, in the example above of a two-node truss restrained at both ends, constraint thermal expansion introduces strain energy that will result in an equivalent increase in the total energy of the model.

Use with other material models

Thermal expansion can be combined with any other (mechanical) material (see "Combining material behaviors," Section 21.1.3) behavior in Abaqus.

Using thermal expansion with other material models

For most materials thermal expansion is defined by a single coefficient or set of orthotropic or anisotropic coefficients or, in Abaqus/Standard, by defining the incremental thermal strains in user subroutine **UEXPAN**. For porous media in Abaqus/Standard, such as soils or rock, thermal expansion can be defined for the solid grains and for the permeating fluid (when using the coupled pore fluid diffusion/stress procedure—see "Coupled pore fluid diffusion and stress analysis," Section 6.8.1). In such a case the thermal expansion definition should be repeated to define the different thermal expansion effects.

Using thermal expansion with gasket behaviors

Thermal expansion can be used in conjunction with any gasket behavior definition. Thermal expansion will affect the expansion of the gasket in the membrane direction and/or the expansion in the gasket's thickness direction.

Elements

Thermal expansion can be used with any stress/displacement or fluid element in Abaqus.

26.1.3 FIELD EXPANSION

Product: Abaqus/Standard

References

- "Material library: overview," Section 21.1.1
- "UEXPAN," Section 1.1.29 of the Abaqus User Subroutines Reference Manual
- *EXPANSION

Overview

Field expansion effects:

- can be defined by specifying field expansion coefficients so that Abaqus/Standard can compute field expansion strains that are driven by changes in predefined field variables;
- can be isotropic, orthotropic, or fully anisotropic;
- are defined as total expansion from a reference value of the predefined field variable;
- can be specified as a function of temperature and/or predefined field variables;
- can be specified directly in user subroutine **UEXPAN** (if the field expansion strains are complicated functions of field variables and state variables); and
- can be defined for more than one predefined field variable.

Defining field expansion coefficients

Field expansion is a material property included in a material definition (see "Material data definition," Section 21.1.2) except when it refers to the expansion of a gasket whose material properties are not defined as part of a material definition. In that case field expansion must be used in conjunction with the gasket behavior definition (see "Defining the gasket behavior directly using a gasket behavior model," Section 32.6.6).

Input File Usage: Use the following options to define field expansion associated with predefined field variable number *n* for most materials:

```
*MATERIAL
*EXPANSION, FIELD=n
```

The *EXPANSION option can be repeated with different values of the predefined field variable number *n* to define field expansion associated with more than one field.

Use the following options to define field expansion associated with predefined field variable number *n* for gaskets whose constitutive response is defined directly as gasket behavior:

∗GASKET BEHAVIOR
∗EXPANSION, FIELD=*n*

The ∗EXPANSION option can be repeated with different values of the predefined field variable number *n* to define field expansion associated with more than one field.

Computation of field expansion strains

Abaqus/Standard requires field expansion coefficients, α_f, that define the total field expansion from a reference value of the predefined field variable *n*, f_n^0, as shown in Figure 26.1.3–1.

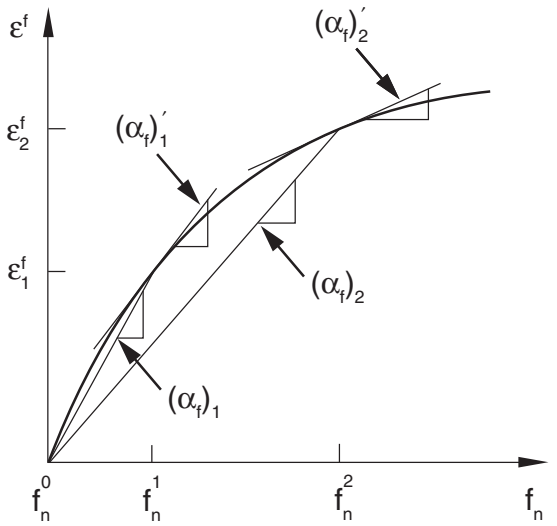


Figure 26.1.3–1 Definition of the field expansion coefficient.

The field expansion for each specified field generates field expansion strains according to the formula

$$\varepsilon^f = \alpha_f(\theta, f_\beta)(f_n - f_n^0) - \alpha_f(\theta^I, f_\beta^I)(f_n^I - f_n^0),$$

where

$\alpha_f(\theta, f_\beta)$	is the field expansion coefficient;
f_n	is the current value of the predefined field variable *n*;
f_n^I	is the initial value of the predefined field variable *n*;
f_β	are the current values of the predefined field variables;

f_β^I are the initial values of the predefined field variables; and

f_n^0 is the reference value of the predefined field variable *n* for the field expansion coefficient.

The second term in the above equation represents the strain due to the difference between the initial value of the predefined field variable*n*, f_n^I, and the corresponding reference value, f_n^0. This term is necessary to enforce the assumption that there is no initial field expansion strain for cases in which the reference value of the predefined field variable *n* does not equal the corresponding initial value.

Defining the reference value of the predefined field variable

If the coefficient of field expansion, α_f, is not a function of temperature or field variables, the reference value of the predefined field variable, f_n^0, is not needed. If α_f is a function of temperature or field variables, you can define f_n^0.

Input File Usage: *EXPANSION, FIELD=*n*, ZERO=f_n^0

Converting field expansion coefficients from differential form to total form

Total field expansion coefficients can be provided directly as outlined in the previous section. However, you may have field expansion data available in differential form:

$$d\varepsilon^f = \alpha_f'(f_n)\, df_n;$$

that is, the tangent to the strain-field variable curve is provided (see Figure 26.1.3–1). To convert to the total field expansion form required by Abaqus, this relationship must be integrated from a suitably chosen reference value of the field variable, f_n^0:

$$\varepsilon^f = \int_{f_n^0}^{f_n} \alpha_f'\, df_n \Rightarrow \alpha_f(f_n) = \frac{1}{f_n - f_n^0} \int_{f_n^0}^{f_n} \alpha_f'\, df_n.$$

For example, suppose α_f' is a series of constant values: $(\alpha_f')_1$ between f_n^0 and f_n^1; $(\alpha_f')_2$ between f_n^1 and f_n^2; $(\alpha')_3$ between f_n^2 and f_n^3; etc. Then,

$$\begin{aligned}
\varepsilon_1^f &= (\alpha_f')_1(f_n^1 - f_n^0),\\
\varepsilon_2^f &= \varepsilon_1^f + (\alpha_f')_2(f_n^2 - f_n^1),\\
\varepsilon_3^f &= \varepsilon_2^f + (\alpha_f')_3(f_n^3 - f_n^2).
\end{aligned}$$

The corresponding total expansion coefficients required by Abaqus are then obtained as

$$\begin{aligned}
(\alpha_f)_1 &= \varepsilon_1^f/(f_n^1 - f_n^0),\\
(\alpha_f)_2 &= \varepsilon_2^f/(f_n^2 - f_n^0),\\
(\alpha_f)_3 &= \varepsilon_3^f/(f_n^3 - f_n^0).
\end{aligned}$$

Defining increments of field expansion strain in user subroutine UEXPAN

Increments of field expansion strain can be specified in user subroutine **UEXPAN** as functions of temperature and/or predefined field variables. User subroutine **UEXPAN** must be used if the field expansion strain increments depend on state variables.

You can use user subroutine **UEXPAN** only once within a single material definition. In particular, you cannot define both thermal and field expansions or multiple field expansions within the same material definition using user subroutine **UEXPAN**.

Input File Usage: *EXPANSION, FIELD=*n*, USER

Defining the initial temperature and field variable values

If the coefficient of field expansion, α_f, is a function of temperature and/or predefined field variables, the initial temperature and initial predefined field variable values, θ^I and f_β^I, are given as described in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1.

Element removal and reactivation

If an element has been removed and subsequently reactivated ("Element and contact pair removal and reactivation," Section 11.2.1), θ^I and f_β^I in the equation for the field expansion strains represent temperature and predefined field variable values as they were at the moment of reactivation.

Defining directionally dependent field expansion

Isotropic, orthotropic, or fully anisotropic field expansion can be defined.

Orthotropic and anisotropic field expansion can be used only with materials where the material directions are defined with local orientations (see "Orientations," Section 2.2.5).

Only isotropic field expansion is allowed with the hyperelastic and hyperfoam material models.

Isotropic expansion

If the field expansion coefficient is defined directly, only one value of α_f is needed at each temperature and/or predefined field variable. If user subroutine **UEXPAN** is used, only one isotropic field expansion strain increment ($\Delta\varepsilon^f = \Delta\varepsilon^f_{11} = \Delta\varepsilon^f_{22} = \Delta\varepsilon^f_{33}$) must be defined.

Input File Usage: Use the following option to define the field expansion coefficient directly:

*EXPANSION, FIELD=*n*, TYPE=ISO

Use the following option to define the field expansion with user subroutine **UEXPAN**:

*EXPANSION, FIELD=*n*, TYPE=ISO, USER

Orthotropic expansion

If the field expansion coefficients are defined directly, the three expansion coefficients in the principal material directions ($\alpha_{f\,11}$, $\alpha_{f\,22}$, and $\alpha_{f\,33}$) should be given as functions of temperature and/or predefined

field variables. If user subroutine **UEXPAN** is used, the three components of field expansion strain increment in the principal material directions ($\Delta\varepsilon^f_{11}$, $\Delta\varepsilon^f_{22}$, and $\Delta\varepsilon^f_{33}$) must be defined.

Input File Usage: Use the following option to define the field expansion coefficients directly:

*EXPANSION, FIELD=*n*, TYPE=ORTHO

Use the following option to define the field expansion with user subroutine **UEXPAN**:

*EXPANSION, FIELD=*n*, TYPE=ORTHO, USER

Anisotropic expansion

If the field expansion coefficients are defined directly, all six components of α_f ($\alpha_{f\,11}$, $\alpha_{f\,22}$, $\alpha_{f\,33}$, $\alpha_{f\,12}$, $\alpha_{f\,13}$, $\alpha_{f\,23}$) must be given as functions of temperature and/or predefined field variables. If user subroutine **UEXPAN** is used, all six components of the field expansion strain increment ($\Delta\varepsilon^f_{11}$, $\Delta\varepsilon^f_{22}$, $\Delta\varepsilon^f_{33}$, $\Delta\varepsilon^f_{12}$, $\Delta\varepsilon^f_{13}$, $\Delta\varepsilon^f_{23}$) must be defined.

Input File Usage: Use the following option to define the field expansion coefficients directly:

*EXPANSION, FIELD=*n*, TYPE=ANISO

Use the following option to define the field expansion with user subroutine **UEXPAN**:

*EXPANSION, FIELD=*n*, TYPE=ANISO, USER

Field expansion stress

When a structure is not free to expand, a change in a predefined field variable will cause stress if there is field expansion associated with that predefined field variable. For example, consider a single 2-node truss of length L that is completely restrained at both ends. The cross-sectional area; the Young's modulus, E; and the field expansion coefficient, α_f, are all constants. The stress in this one-dimensional problem can then be calculated from Hooke's Law as $\sigma_x = E(\varepsilon_x - \varepsilon^f_x)$, where ε_x is the total strain and $\varepsilon^f_x = \alpha_f \Delta f_n$ is the field expansion strain, where Δf_n is the change in the value of the predefined field variable number n. Since the element is fully restrained, $\varepsilon_x = 0$. If the values of the field variable at both nodes are the same, we obtain the stress $\sigma_x = -E\alpha_f \Delta f_n$.

Depending on the value of the field expansion coefficient and the change in the value of the corresponding predefined field variable, a constrained field expansion can cause significant stress and introduce strain energy that will result in an equivalent increase in the total energy of the model. Therefore, it is often important to define boundary conditions with particular care for problems involving this property to avoid overconstraining the field expansion.

Use with other material models

Field expansion can be combined with any other (mechanical) material (see "Combining material behaviors," Section 21.1.3) behavior in Abaqus/Standard.

Using field expansion with other material models

For most materials field expansion is defined by a single coefficient or a set of orthotropic or anisotropic coefficients or by defining the incremental field expansion strains in user subroutine **UEXPAN**.

Using field expansion with gasket behavior

Field expansion can be used in conjunction with any gasket behavior definition. Field expansion will affect the expansion of the gasket in the membrane direction and/or the expansion in the gasket's thickness direction.

Elements

Field expansion can be used with any stress/displacement element in Abaqus/Standard, except for beam and shell elements using a general section behavior.

26.1.4 VISCOSITY

Products: Abaqus/Explicit Abaqus/CFD Abaqus/CAE

References

- "Viscous shear behavior" in "Equation of state," Section 25.2.1
- *VISCOSITY
- *EOS
- *TRS
- "Defining viscosity" in "Defining other mechanical models," Section 12.9.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Material shear viscosity is an internal property of a fluid that offers resistance to flow. It can be specified in Abaqus/Explicit and Abaqus/CFD.

Material shear viscosity in Abaqus/Explicit:

- can be a function of temperature and shear strain rate; and
- must be used in combination with an equation of state ("Equation of state," Section 25.2.1).

Material shear viscosity in Abaqus/CFD:

- can be a function of temperature only for the Newtonian model;
- can be a function of shear strain rate; and
- is not supported for field-dependent variants.

Viscous shear behavior

The resistance to flow of a viscous fluid is described by the following relationship between deviatoric stress and strain rate

$$\mathbf{S} = 2\eta\dot{\mathbf{e}} = \eta\dot{\gamma},$$

where $\mathbf{S}$ is the deviatoric stress, $\dot{\mathbf{e}}$ is the deviatoric part of the strain rate, η is the viscosity, and $\dot{\gamma} = 2\dot{\mathbf{e}}$ is the engineering shear strain rate.

Newtonian fluids are characterized by a viscosity that only depends on temperature, $\eta(\theta)$. In the more general case of non-Newtonian fluids the viscosity is a function of the temperature and shear strain rate:

$$\eta = \eta(\dot{\gamma}, \theta),$$

where $\dot{\gamma} = \sqrt{\frac{1}{2}\dot{\boldsymbol{\gamma}} : \dot{\boldsymbol{\gamma}}}$ is the equivalent shear strain rate. In terms of the equivalent shear stress, $\tau = \sqrt{\frac{1}{2}\mathbf{S} : \mathbf{S}}$, we have:

$$\tau = \eta\dot{\gamma}.$$

Non-Newtonian fluids can be classified as shear-thinning (or pseudoplastic), when the apparent viscosity decreases with increasing shear rate, and shear-thickening (or dilatant), when the viscosity increases with strain rate.

In addition to the Newtonian viscous fluid model, Abaqus/CFD and Abaqus/Explicit support several models of nonlinear viscosity to describe non-Newtonian fluids: power law, Carreau-Yasuda, Cross, Herschel-Bulkley, Powell-Eyring, and Ellis-Meter. Other functional forms of the viscosity can also be specified in tabular format. In addition, in Abaqus/Explicit user subroutine **`VUVISCOSITY`** can be used.

Newtonian

The Newtonian model is useful to model viscous laminar flow governed by the Navier-Poisson law of a Newtonian fluid, $\tau = \eta\dot{\gamma}$. Newtonian fluids are characterized by a viscosity that depends only on temperature, $\eta(\theta)$. You need to specify the viscosity as a tabular function of temperature when you define the Newtonian viscous deviatoric behavior.

Input File Usage: ∗VISCOSITY, DEFINITION=NEWTONIAN (default)

Abaqus/CAE Usage: Property module: material editor: **Mechanical→Viscosity**

Power law

The power law model is commonly used to describe the viscosity of non-Newtonian fluids. The viscosity is expressed as

$$\eta = k\dot{\gamma}^{n-1}; \quad \eta_{\min} \le \eta \le \eta_{\max},$$

where k is the flow consistency index and n is the flow behavior index. When $n < 1$, the fluid is shear-thinning (or pseudoplastic): the apparent viscosity decreases with increasing shear rate. When $n > 1$, the fluid is shear-thickening (or dilatant); and when $n = 1$, the fluid is Newtonian. Optionally, you can place a lower limit, $\eta_{\min}$, and/or an upper limit, $\eta_{\max}$, on the value of the viscosity computed from the power law.

Input File Usage: ∗VISCOSITY, DEFINITION=POWER LAW

Abaqus/CAE Usage: The power law model is not supported in Abaqus/CAE.

Carreau-Yasuda

The Carreau-Yasuda model describes the shear thinning behavior of polymers. This model often provides a better fit than the power law model for both high and low shear strain rates. The viscosity is expressed as

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty)\left(1 + (\lambda\dot{\gamma})^a\right)^{\frac{n-1}{a}},$$

where η_0 is the low shear rate Newtonian viscosity, η_∞ is the infinite shear viscosity (at high shear strain rates), λ is the natural time constant of the fluid ($1/\lambda$ is the critical shear rate at which the fluid changes from Newtonian to power law behavior), and n represents the flow behavior index in the power law regime. The coefficient a is a material parameter. The original Carreau model is recovered when a=2.

Input File Usage: *VISCOSITY, DEFINITION=CARREAU-YASUDA

Abaqus/CAE Usage: The Carreau-Yasuda model is not supported in Abaqus/CAE.

Cross

The Cross model is commonly used when it is necessary to describe the low-shear-rate behavior of the viscosity. The viscosity is expressed as

$$\eta = \eta_\infty + \frac{(\eta_0 - \eta_\infty)}{1 + (\lambda\dot{\gamma})^{1-n}},$$

where η_0 is the Newtonian viscosity, η_∞ is the infinite shear viscosity (usually assumed to be zero for the Cross model), λ is the natural time constant of the fluid ($1/\lambda$ is the critical shear rate at which the fluid changes from Newtonian to power-law behavior), and n is the flow behavior index in the power law regime.

Input File Usage: *VISCOSITY, DEFINITION=CROSS

Abaqus/CAE Usage: The Cross model is not supported in Abaqus/CAE.

Herschel-Bulkley

The Herschel-Bulkley model can be used to describe the behavior of viscoplastic fluids, such as Bingham plastics, that exhibit a yield response. The viscosity is expressed as

$$\eta = \begin{cases} \eta_0 & \text{if} \quad \tau < \tau_0; \\ \frac{1}{\dot{\gamma}}\left(\tau_0 + k(\dot{\gamma}^n - (\tau_0/\eta_0)^n)\right) & \text{if} \quad \tau \geq \tau_0. \end{cases}$$

Here τ_0 is the "yield" stress and η_0 is a penalty viscosity to model the "rigid-like" behavior in the very low strain rate regime ($\dot{\gamma} \leq \tau_0/\eta_0$), when the stress is below the yield stress, $\tau \leq \tau_0$. With increasing strain rates, the viscosity transitions into a power law model once the yield threshold is reached, $\tau \geq \tau_0$. The parameters k and n are the flow consistency and the flow behavior indexes in the power law regime, respectively. Bingham plastics correspond to the case n=1.

Input File Usage: *VISCOSITY, DEFINITION=HERSCHEL-BULKLEY

Abaqus/CAE Usage: The Herschel-Bulkley model is not supported in Abaqus/CAE.

Powell-Eyring

This model, which is derived from the theory of rate processes, is relevant primarily to molecular fluids but can be used in some cases to describe the viscous behavior of polymer solutions and viscoelastic suspensions over a wide range of shear rates. The viscosity is expressed as

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty)\frac{\sinh^{-1}(\lambda\dot{\gamma})}{\lambda\dot{\gamma}},$$

where η_0 is the Newtonian viscosity, η_∞ is the infinite shear viscosity, and λ represents a characteristic time of the measured system.

Input File Usage: *VISCOSITY, DEFINITION=POWELL-EYRING

Abaqus/CAE Usage: The Powell-Eyring model is not supported in Abaqus/CAE.

Ellis-Meter

The Ellis-Meter model expresses the viscosity in terms of the effective shear stress, $\tau = \sqrt{\frac{1}{2}\mathbf{S} : \mathbf{S}}$, as:

$$\eta = \eta_\infty + \frac{(\eta_0 - \eta_\infty)}{1 + (\tau/\tau_{1/2})^{(1-n)/n}},$$

where $\tau_{1/2}$ is the effective shear stress at which the viscosity is 50% between the Newtonian limit, η_0, and the infinite shear viscosity, η_∞, and n represents the flow index in the power law regime.

Input File Usage: *VISCOSITY, DEFINITION=ELLIS-METER

Abaqus/CAE Usage: The Ellis-Meter model is not supported in Abaqus/CAE.

Tabular

In Abaqus/Explicit the viscosity can be specified directly as a tabular function of shear strain rate and temperature. In Abaqus/CFD only shear strain rate dependence is supported.

Input File Usage: *VISCOSITY, DEFINITION=TABULAR

Abaqus/CAE Usage: Specifying the viscosity directly as a tabular function is not supported in Abaqus/CAE.

User-defined (Abaqus/Explicit only)

In Abaqus/Explicit you can specify a user-defined viscosity in user subroutine **VUVISCOSITY** (see "VUVISCOSITY," Section 1.2.21 of the Abaqus User Subroutines Reference Manual).

Input File Usage: *VISCOSITY, DEFINITION=USER

Abaqus/CAE Usage: User-defined viscosity is not supported in Abaqus/CAE.

Temperature dependence of viscosity (Abaqus/Explicit only)

The temperature-dependence of the viscosity of many polymer materials of industrial interest obeys a time-temperature shift relationship in the form:

$$\eta(\dot{\gamma}, \theta) = a_T(\theta)\eta(a_T(\theta)\dot{\gamma}, \theta_0),$$

where $a_T(\theta)$ is the shift function and θ_0 is the reference temperature at which the viscosity versus shear strain rate relationship is known. This concept for temperature dependence is usually referred to as thermo-rheologically simple (TRS) temperature dependence. In the Newtonian limit for low shear rates, when $\dot{\gamma} \to 0$, we have

$$\eta_0(\theta) = \lim_{\dot{\gamma}\to 0} \eta(\dot{\gamma}\theta) = a_T(\theta)\eta_0(\theta_0).$$

Thus, the shift function is defined as the ratio of the Newtonian viscosity at the temperature of interest to that of the chosen reference state: $a_T(\theta) = \eta_0(\theta)/\eta_0(\theta_0)$.

See "Thermo-rheologically simple temperature effects" in "Time domain viscoelasticity," Section 22.7.1, for a description of the different forms of the shift function available in Abaqus.

Input File Usage: Use the following options to define a thermo-rheologically simple (TRS) temperature-dependent viscosity:

∗VISCOSITY
∗TRS

Abaqus/CAE Usage: Defining a thermo-rheologically simple temperature-dependent viscosity is not supported in Abaqus/CAE.

Use with other material models

Material shear viscosity in Abaqus/Explicit must be used in combination with an equation of state to define the material's volumetric mechanical behavior (see "Equation of state," Section 25.2.1).

Elements

Material shear viscosity can be used with any solid (continuum) elements in Abaqus/Explicit except plane stress elements and with any fluid (continuum) elements in Abaqus/CFD.

26.2 Heat transfer properties

- "Thermal properties: overview," Section 26.2.1
- "Conductivity," Section 26.2.2
- "Specific heat," Section 26.2.3
- "Latent heat," Section 26.2.4

26.2.1 THERMAL PROPERTIES: OVERVIEW

The following properties describe the thermal behavior of a material and can be used in heat transfer and thermal stress analyses (see "Heat transfer analysis procedures: overview," Section 6.5.1):

- **Thermal conductivity:** When heat flows by conduction, the thermal conductivity must be defined ("Conductivity," Section 26.2.2).
- **Specific heat:** In transient heat transfer analyses as well as adiabatic stress analyses the specific heat of a material must be defined ("Specific heat," Section 26.2.3).
- **Latent heat:** When a material changes phase, the change in internal energy can be significant. The amount of energy liberated or absorbed can be defined by specifying a latent heat for each phase change a material undergoes ("Latent heat," Section 26.2.4).

26.2.2 CONDUCTIVITY

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Thermal properties: overview," Section 26.2.1
- *CONDUCTIVITY
- "Specifying thermal conductivity," Section 12.10.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A material's thermal conductivity:

- must be defined for "Uncoupled heat transfer analysis," Section 6.5.2; "Fully coupled thermal-stress analysis," Section 6.5.3; and "Coupled thermal-electrical analysis," Section 6.7.3;
- must be defined for an Abaqus/CFD analysis when the energy equation is active ("Energy equation" in "Incompressible fluid dynamic analysis," Section 6.6.2);
- can be linear or nonlinear (by defining it as a function of temperature);
- can be isotropic, orthotropic, or fully anisotropic; and
- can be specified as a function of temperature and/or field variables.

Directional dependence of thermal conductivity

Isotropic, orthotropic, or fully anisotropic thermal conductivity can be defined. Only isotropic thermal conductivity can be defined for an incompressible fluid dynamic analysis that includes an energy equation. For orthotropic or anisotropic thermal conductivity, a local orientation ("Orientations," Section 2.2.5) must be used to specify the material directions used to define the conductivity.

Isotropic conductivity

For isotropic conductivity only one value of conductivity is needed at each temperature and field variable value. Isotropic conductivity is the default.

Input File Usage: *CONDUCTIVITY, TYPE=ISO

Abaqus/CAE Usage: Property module: material editor: **Thermal→Conductivity**: **Type: Isotropic**

Orthotropic conductivity

For orthotropic conductivity three values of conductivity (k_{11}, k_{22}, k_{33}) are needed at each temperature and field variable value.

Input File Usage: *CONDUCTIVITY, TYPE=ORTHO

Abaqus/CAE Usage: Property module: material editor: **Thermal→Conductivity**:
Type: Orthotropic

Anisotropic conductivity

For fully anisotropic conductivity six values of conductivity (k_{11}, k_{12}, k_{22}, k_{13}, k_{23}, k_{33}) are needed at each temperature and field variable value.

Input File Usage: *CONDUCTIVITY, TYPE=ANISO

Abaqus/CAE Usage: Property module: material editor: **Thermal→Conductivity**:
Type: Anisotropic

Elements

Thermal conductivity is active in all heat transfer, coupled temperature-displacement, coupled thermal-electrical-structural, and coupled thermal-electrical elements in Abaqus. Isotropic thermal conductivity is active in fluid (continuum) elements in Abaqus/CFD for incompressible fluid dynamic analyses that include an energy equation.

26.2.3 SPECIFIC HEAT

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CFD Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Thermal properties: overview," Section 26.2.1
- *SPECIFIC HEAT
- "Defining specific heat," Section 12.10.6 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A material's specific heat:

- is required for transient "Uncoupled heat transfer analysis," Section 6.5.2; transient "Fully coupled thermal-stress analysis," Section 6.5.3; transient "Coupled thermal-electrical analysis," Section 6.7.3; and "Adiabatic analysis," Section 6.5.4;
- must be defined for an Abaqus/CFD analysis when the energy equation is active ("Energy equation" in "Incompressible fluid dynamic analysis," Section 6.6.2);
- must appear in conjunction with a density definition (see "Density," Section 21.2.1);
- can be linear or nonlinear (by defining it as a function of temperature); and
- can be specified as a function of temperature and/or field variables.

Defining specific heat

The specific heat of a substance is defined as the amount of heat required to increase the temperature of a unit mass by one degree. Mathematically, this physical statement can be expressed as:

$$c = \frac{\delta Q}{d\theta} = \theta\left(\frac{ds}{d\theta}\right),$$

where δQ is the infinitessimal heat added per unit mass and s is the entropy per unit mass. Since heat transfer depends on the conditions encountered during the whole process (a path function), it is necessary to specify the conditions used in the process to unambiguously characterize the specific heat. Thus, a process where the heat is supplied keeping the volume constant defines the specific heat as:

$$c_v = \left(\frac{\delta Q}{d\theta}\right)\Big|_v = \theta\left(\frac{\partial s}{\partial \theta}\right)\Big|_v = \left(\frac{\partial u}{\partial \theta}\right)\Big|_v,$$

where u is the internal energy per unit mass.

Whereas, a process where the heat is supplied keeping the pressure constant defines the specific heat as:

$$c_p = \left(\frac{\delta Q}{d\theta}\right)\Big|_p = \theta\left(\frac{\partial s}{\partial \theta}\right)\Big|_p = \left(\frac{\partial h}{\partial \theta}\right)\Big|_p,$$

where $h = u + pv$ is the enthalpy per unit mass. In general, the specific heats are functions of temperature. For solids and liquids, c_v and c_p are equivalent; thus, there is no need to distinguish between them. When possible, large changes in internal energy or enthalpy during a phase change should be modeled using "Latent heat," Section 26.2.4, instead of specific heat.

Defining constant-volume specific heat

The specific heat per unit mass is given as a function of temperature and field variables. By default, specific heat at constant volume is assumed.

Input File Usage: *SPECIFIC HEAT

The following option can also be used in Abaqus/CFD:

*SPECIFIC HEAT, TYPE=CONSTANT VOLUME

Abaqus/CAE Usage: Property module: material editor: **Thermal→Specific Heat**; **Type**: **Constant Volume**

Defining constant-pressure specific heat

In Abaqus/CFD the constant-pressure specific heat is required when the energy equation is used for thermal-flow problems.

Input File Usage: *SPECIFIC HEAT, TYPE=CONSTANT PRESSURE

Abaqus/CAE Usage: Property module: material editor: **Thermal→Specific Heat**; **Type**: **Constant Pressure**

Elements

Specific heat effects can be defined for all heat transfer, coupled thermal-electrical-structural, coupled temperature-displacement, coupled thermal-electrical, and fluid elements in Abaqus. Specific heat can also be defined for stress/displacement elements for use in adiabatic stress analysis.

Specific heat must be defined for all transient thermal analyses even if the only elements in the model are user-defined elements ("User-defined elements," Section 32.15.1), in which case a dummy specific heat must be specified.

26.2.4 LATENT HEAT

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- "Thermal properties: overview," Section 26.2.1
- *LATENT HEAT
- "Specifying latent heat data," Section 12.10.5 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A material's latent heat:

- models large changes in internal energy during phase change of a material;
- is active only during transient heat transfer, coupled thermal-stress, coupled thermal-electrical-structural and coupled thermal-electrical analysis in Abaqus (see "Heat transfer analysis procedures: overview," Section 6.5.1);
- must appear in conjunction with a density definition (see "Density," Section 21.2.1); and
- always makes an analysis nonlinear.

Defining latent heat

Latent heat effects can be significant and must be included in many heat transfer problems involving phase change. When latent heat is given, it is assumed to be in addition to the specific heat effect (see "Uncoupled heat transfer analysis," Section 2.11.1 of the Abaqus Theory Manual, for details).

The latent heat is assumed to be released over a range of temperatures from a lower (solidus) temperature to an upper (liquidus) temperature. To model a pure material with a single phase change temperature, these limits can be made very close.

As many latent heats as are necessary can be defined to model several phase changes in the material. Latent heat can be combined with any other material behavior in Abaqus, but it should not be included in the material definition unless necessary; it always makes the analysis nonlinear.

Direct data specification

If the phase change occurs within a known temperature range, the solidus and liquidus temperatures can be given directly. The latent heat should be given per unit mass.

Input File Usage: *LATENT HEAT

Abaqus/CAE Usage: Property module: material editor: **Thermal→Latent Heat**

User subroutine

In some cases it may be necessary to include a kinetic theory for the phase change to model the effect accurately in Abaqus/Standard; for example, the prediction of crystallization in a polymer casting process. In such cases you can model the process in considerable detail using solution-dependent state variables ("User subroutines: overview," Section 18.1.1) and user subroutine **HETVAL**.

Input File Usage: Use the following options:

*HEAT GENERATION
*DEPVAR

Abaqus/CAE Usage: Property module: material editor:
Thermal→Heat Generation
General→Depvar

Elements

Latent heat effects can be used in all diffusive heat transfer, coupled temperature-displacement, coupled thermal-electrical-structural and coupled thermal-electrical elements in Abaqus but cannot be used with convective heat transfer elements. Strong latent heat effects are best modeled with first-order or modified second-order elements, which use integration methods designed to provide accurate results for such cases.

See "Freezing of a square solid: the two-dimensional Stefan problem," Section 1.6.2 of the Abaqus Benchmarks Manual, for an example of a heat conduction problem involving latent heat.

26.3 Acoustic properties

- “Acoustic medium,” Section 26.3.1

26.3.1 ACOUSTIC MEDIUM

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1
- "Acoustic and shock loads," Section 33.4.6
- "Material library: overview," Section 21.1.1
- "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1
- *ACOUSTIC MEDIUM
- *DENSITY
- *INITIAL CONDITIONS
- "Defining an acoustic medium," Section 12.12.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

An acoustic medium:

- is used to model sound propagation problems;
- can be used in a purely acoustic analysis or in a coupled acoustic-structural analysis such as the calculation of shock waves in a fluid or noise levels in a vibration problem;
- is an elastic medium (usually a fluid) in which stress is purely hydrostatic (no shear stress) and pressure is proportional to volumetric strain;
- is specified as part of a material definition;
- must appear in conjunction with a density definition (see "Density," Section 21.2.1);
- can include fluid cavitation in Abaqus/Explicit when the absolute pressure drops to a limit value;
- can be defined as a function of temperature and/or field variables;
- can include dissipative effects;
- can model small pressure changes (small amplitude excitation);
- can model waves in the presence of steady underlying flow of the medium; and
- is active only during dynamic analysis procedures ("Dynamic analysis procedures: overview," Section 6.3.1).

Defining an acoustic medium

The equilibrium equation for small motions of a compressible, inviscid fluid flowing through a resisting matrix material is taken to be

$$\frac{\partial p}{\partial \mathbf{x}} + \gamma \dot{\mathbf{u}}^f + \rho_f \ddot{\mathbf{u}}^f = 0,$$

where p is the dynamic pressure in the fluid (the pressure in excess of any initial static pressure), $\mathbf{x}$ is the spatial position of the fluid particle, $\dot{\mathbf{u}}^f$ is the fluid particle velocity, $\ddot{\mathbf{u}}^f$ is the fluid particle acceleration, ρ_f is the density of the fluid, and γ is the "volumetric drag" (force per unit volume per velocity) caused by the fluid flowing through the matrix material. The d'Alembert term has been written without convection on the assumption that there is no steady flow of the fluid, which is usually considered to be sufficiently accurate for steady fluid velocities up to Mach 0.1.

The constitutive behavior of the fluid is assumed to be inviscid and compressible, so that the bulk modulus of an acoustic medium relates the dynamic pressure in the medium to the volumetric strain by

$$p = -\mathrm{K}_f \varepsilon_V,$$

where $\varepsilon_V = \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}$ is the volumetric strain. Both the bulk modulus K_f and the density ρ_f of an acoustic medium must be defined.

The bulk modulus K_f can be defined as a function of temperature and field variables but does not vary in value during an implicit dynamic analysis using the subspace projection method ("Implicit dynamic analysis using direct integration," Section 6.3.2) or a direct-solution steady-state dynamic analysis ("Direct-solution steady-state dynamic analysis," Section 6.3.4); for these procedures the value of the bulk modulus at the beginning of the step is used.

Input File Usage: Use both of the following options to define an acoustic medium:

*ACOUSTIC MEDIUM, BULK MODULUS
*DENSITY

Abaqus/CAE Usage: Property module: material editor:
Other→Acoustic Medium: **Bulk Modulus**
General→Density

Volumetric drag

Dissipation of energy (and attenuation of acoustic waves) may occur in an acoustic medium due to a variety of factors. Such dissipation effects are phenomenologically characterized in the frequency domain by the imaginary part of the propagation constant, which gives an exponential decay in amplitude as a function of distance. In Abaqus the simplest way to model this effect is through a "volumetric drag coefficient," γ (force per unit volume per velocity).

In frequency-domain procedures, γ may be frequency dependent. γ can be entered as a function of frequency—$\gamma(f)$, where f is the frequency in cycles per time (usually Hz)—in addition to temperature and/or field variables only when the acoustic medium is used in a steady-state dynamics procedure. If the acoustic medium is used in a direct-integration dynamic procedure (including Abaqus/Explicit), the volumetric drag coefficient is assumed to be independent of frequency and the first value entered for the current temperature and/or field variable is used.

In all procedures except direct steady-state dynamics the gradient of γ/ρ_f is assumed to be small.

Input File Usage: *ACOUSTIC MEDIUM, VOLUMETRIC DRAG

Abaqus/CAE Usage: Property module: material editor: **Other→Acoustic Medium**: **Volumetric Drag**: **Include volumetric drag**

Porous acoustic material models

Porous materials are commonly used to suppress acoustic waves; this attenuating effect arises from a number of effects as the acoustic fluid interacts with the solid matrix. For many categories of materials, the solid matrix can be approximated as either fully rigid compared to the acoustic fluid or fully limp. In these cases a mechanical model that resolves only acoustic waves will suffice. The acoustic behavior of porous materials can be modeled in a variety of ways in Abaqus/Standard.

Craggs model

The model discussed in Craggs (1978) is readily accommodated in Abaqus. Applying this model results in the dynamic equilibrium equation for the fluid expressed in this form:

$$\nabla^2\tilde{p} + (ka)^2 K_s \Omega \tilde{p} - ika\frac{R\Omega}{\rho_f c_f} = 0,$$

where R is the real-valued resistivity, Ω is the real-valued dimensionless porosity, K_s is the dimensionless "structure factor," and $ka = \frac{\omega}{c_f}$ is the dimensionless wave number. This equation can be rewritten as

$$\nabla^2\tilde{p} + \frac{(\omega)^2}{\rho_f c_f^2}\left[\rho_f K_s \Omega + \frac{R\Omega}{i\omega}\right] = 0.$$

This model, therefore, can be applied straightforwardly in Abaqus by setting the material density equal to $\rho_f K_s \Omega$, the volumetric drag equal to $R\Omega$, and the bulk modulus equal to $\rho_f c_f^2$. The Craggs model is supported for all acoustic procedures in Abaqus.

Delany-Bazley and Delany-Bazley-Miki models

Abaqus/Standard supports the well-known empirical model proposed in Delany & Bazley (1970), which determines the material properties as a function of frequency and user-defined flow resistivity, R; density, ρ_f; and bulk modulus, K_f. A variation on this model, proposed by Miki (1990) is also available. These models are supported only for steady-state dynamic procedures.

Both models compute frequency-dependent material characteristic impedance, $\tilde{Z}$, and wavenumber or propagation constant, $\tilde{k}$, according to the following formula:

$$\tilde{k} \equiv \frac{\omega}{c_0}\left(1 + C_1 x^{C_2} - iC_3 x^{C_4}\right),$$

$$\tilde{Z} \equiv \rho_f c_0\left(1 + C_5 x^{C_6} - iC_7 x^{C_8}\right),$$

where

$$c_0 \equiv \sqrt{\frac{K_f}{\rho_f}},$$

and

$$x \equiv \frac{\rho_f \omega}{2\pi R}.$$

The constants are as given in the table below:

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Delany-Bazley	0.0978	–0.7	0.189	–0.595	0.0571	–0.754	0.087	–0.732
Miki	0.1227	–0.618	0.1792	–0.618	0.0786	–0.632	0.1205	–0.632

The material characteristic impedance and the wavenumber are converted internally to complex density and complex bulk modulus for use in Abaqus. The signs of the imaginary parts in these formulae are consistent with the Abaqus sign convention for time-harmonic dynamics.

Input File Usage: Use the following options to use the Delany-Bazley model:
*DENSITY
*ACOUSTIC MEDIUM, BULK MODULUS
*ACOUSTIC MEDIUM, POROUS MODEL=DELANY BAZLEY
Use the following options to use the Miki model:
*DENSITY
*ACOUSTIC MEDIUM, BULK MODULUS
*ACOUSTIC MEDIUM, POROUS MODEL=MIKI

Abaqus/CAE Usage: Porous acoustic material models are not supported in Abaqus/CAE.

General frequency-dependent models

For steady-state dynamic procedures, Abaqus/Standard supports general frequency-dependent complex bulk modulus and complex density. Using these parameters, data from a wide range of models can be accommodated in an analysis; for example, see Allard, et. al (1998), Attenborough (1982), Song & Bolton (1999), and Wilson (1993). These models are used in a variety of applications, such as ocean acoustics, aerospace, automotive, and architectural acoustic engineering.

The signs of these parameters must be consistent with the sign conventions used in Abaqus, and with conservation of energy. Abaqus uses a Fourier transform pair formally equivalent to assuming $e^{i\omega t}$ time dependence. Consequently, the real parts of the density and bulk modulus are positive for all values of frequency, the imaginary part of the bulk modulus must be positive, and the imaginary part of the density must be negative.

A linear isotropic acoustic material can be fully described with the two frequency-dependent parameters: bulk modulus, $\tilde{K}_f$, and density, $\tilde{\rho}_f$. It is common, however, to encounter materials defined in terms of other parameter pairs, such as characteristic impedance, $\tilde{Z}$, wave number or propagation

constant, $\tilde{k}$, or speed of sound, $\tilde{c}$. Data defined with the pair $\left(\tilde{Z},\tilde{k}\right)$ or $\left(\tilde{Z},\tilde{c}\right)$ can be converted into the complex density and bulk modulus form, beginning from the following standard formulae:

$$\tilde{Z} \equiv \sqrt{\tilde{\rho}_f \tilde{K}_f},$$

$$\tilde{k} \equiv \omega \sqrt{\frac{\tilde{\rho}_f}{\tilde{K}_f}},$$

$$\tilde{c} \equiv \sqrt{\frac{\tilde{\rho}_f}{\tilde{K}_f}}.$$

Consistent with the Abaqus sign conventions, the real parts of $\tilde{k}$ and $\tilde{c}$ must be positive; the imaginary part of $\tilde{k}$ must be negative, and the imaginary part of $\tilde{c}$ must be positive. In commonly observed materials, the ratio of the magnitude of the imaginary part to the real part for each of these constants is usually much less than one.

Input File Usage: Use the following option to use the general frequency-dependent model:
*ACOUSTIC MEDIUM, COMPLEX BULK MODULUS
*ACOUSTIC MEDIUM, COMPLEX DENSITY

If desired, either complex material option can be used instead in conjunction with a real-valued, frequency-independent material option:

*ACOUSTIC MEDIUM, COMPLEX BULK MODULUS
*DENSITY

or, alternatively,

*ACOUSTIC MEDIUM, BULK MODULUS
*ACOUSTIC MEDIUM, COMPLEX DENSITY

Abaqus/CAE Usage: General frequency-dependent acoustic material models are not supported in Abaqus/CAE.

Conversion from complex material impedance and wavenumber

Since

$$\tilde{\rho}_f = \frac{1}{\omega}\tilde{k}\tilde{Z}$$

and

$$\tilde{K}_f = \omega \frac{\tilde{Z}}{\tilde{k}},$$

the real and imaginary parts of $\tilde{\rho}_f$ are, respectively:

$$\Re(\tilde{\rho}_f) = \frac{1}{\omega}\big(\Re(\tilde{k})\Re(\tilde{Z}) - \Im(\tilde{k})\Im(\tilde{Z})\big),$$

$$\Im(\tilde{\rho}_f) = \frac{1}{\omega}\big(\Im(\tilde{k})\Re(\tilde{Z}) + \Re(\tilde{k})\Im(\tilde{Z})\big);$$

and the real and imaginary parts of $\tilde{K}_f$ are, respectively:

$$\Re(\tilde{K}_f) = \frac{\omega}{|\tilde{k}|^2}\big(\Re(\tilde{k})\Re(\tilde{Z}) + \Im(\tilde{k})\Im(\tilde{Z})\big),$$

$$\Im(\tilde{K}_f) = \frac{\omega}{|\tilde{k}|^2}\big(\Re(\tilde{k})\Im(\tilde{Z}) - \Im(\tilde{k})\Re(\tilde{Z})\big).$$

.

Conversion from complex impedance and speed of sound

Since

$$\tilde{\rho}_f = \frac{\tilde{Z}}{\tilde{c}}$$

and

$$\tilde{K}_f = \tilde{Z}\tilde{c},$$

the real and imaginary parts of $\tilde{\rho}_f$ are, respectively:

$$\Re(\tilde{\rho}_f) = \frac{1}{|\tilde{c}|^2}\big(\Re(\tilde{c})\Re(\tilde{Z}) + \Im(\tilde{c})\Im(\tilde{Z})\big),$$

$$\Im(\tilde{\rho}_f) = \frac{1}{|\tilde{c}|^2}\big(\Re(\tilde{c})\Im(\tilde{Z}) - \Im(\tilde{c})\Re(\tilde{Z})\big);$$

and the real and imaginary parts of $\tilde{K}_f$ are, respectively:

$$\Re(\tilde{K}_f) = \big(\Re(\tilde{c})\Re(\tilde{Z}) - \Im(\tilde{c})\Im(\tilde{Z})\big),$$

$$\Im(\tilde{K}_f) = \big(\Im(\tilde{c})\Re(\tilde{Z}) + \Re(\tilde{c})\Im(\tilde{Z})\big).$$

.

Fluid cavitation

In general, fluids cannot withstand any significant tensile stress and are likely to undergo large volume expansion when the absolute pressure is close to or less than zero. Abaqus/Explicit allows modeling of this phenomenon through a cavitation pressure limit for the acoustic medium. When the fluid absolute pressure (sum of the dynamic and initial static pressures) reduces to this limit, the fluid undergoes free volume expansion (i.e., cavitation), without a further drop in the pressure. If this limit is not defined, the fluid is assumed not to undergo cavitation even under a tensile, negative absolute pressure, condition.

The constitutive behavior for an acoustic medium capable of undergoing cavitation can be stated as

$$p = \max\left\{p_v,\ p_c - p_0\right\},$$

where a pseudo-pressure p_v, a measure of the volumetric strain, is defined as

$$p_v = -\mathrm{K}_f \varepsilon_V,$$

where p_c is the fluid cavitation limit and p_0 is the initial acoustic static pressure. A total wave formulation is used for a nonlinear acoustic medium undergoing cavitation. This formulation is very similar to the scattered wave formulation except that the pseudo-pressure, defined as the product of the bulk modulus and the compressive volumetric strain, plays the role of the material state variable instead of the acoustic dynamic pressure and the acoustic dynamic pressure is readily available from this pseudo-pressure subject to the cavitation condition.

Input File Usage: *ACOUSTIC MEDIUM, CAVITATION LIMIT

Abaqus/CAE Usage: Fluid cavitation is not supported in Abaqus/CAE.

Defining the wave formulation

In the presence of cavitation in Abaqus/Explicit the fluid mechanical behavior is nonlinear. Hence, for an acoustic problem with incident wave loading and possible cavitation in the fluid, the scattered wave formulation, which provides a solution for only a scattered wave dynamic acoustic pressure, may not be appropriate. For these cases the total wave formulation, which solves for the total dynamic acoustic pressure, should be selected. See "Acoustic and shock loads," Section 33.4.6, for details.

Input File Usage: *ACOUSTIC WAVE FORMULATION, TYPE=TOTAL WAVE

Abaqus/CAE Usage: Any module: **Model→Edit Attributes→*model_name***. Toggle on **Specify acoustic wave formulation**: **Total wave**

Defining the initial acoustic static pressure

Cavitation occurs when the absolute pressure reaches the cavitation limit value. Abaqus/Explicit allows for an initial linearly varying hydrostatic pressure in the fluid medium (see "Defining initial acoustic static pressure" in "Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1). You can specify pressure values at two locations and a node set of the acoustic medium nodes. Abaqus/Explicit interpolates from these data to initialize the static pressure at all the nodes in the specified node set. If the

pressure at only one location is specified, the hydrostatic pressure in the fluid is assumed to be uniform. The acoustic static pressure is used only for determining the cavitation status of the acoustic element nodes and does not apply any static loads to the acoustic or structural mesh at their common wetted interface.

Input File Usage: *INITIAL CONDITIONS, TYPE=ACOUSTIC STATIC PRESSURE

Abaqus/CAE Usage: Initial acoustic pressures are not supported in Abaqus/CAE.

Defining a steady flow field

Acoustic finite elements can be used to simulate time-harmonic wave propagation and natural frequency analysis in the presence of a steady mean flow of the medium. For example, air may move at a speed large enough to affect the propagation speed of waves in the direction of flow and against it. These effects are modeled in Abaqus/Standard by specifying an acoustic flow velocity during the linear perturbation analysis step definition; you do not need to alter the acoustic material properties. See "Acoustic, shock, and coupled acoustic-structural analysis," Section 6.10.1, for details.

Elements

An acoustic material definition can be used only with the acoustic elements in Abaqus (see "Choosing the appropriate element for an analysis type," Section 27.1.3).

In Abaqus/Standard second-order acoustic elements are more accurate than first-order elements. Use at least six nodes per wavelength λ in the acoustic medium to obtain accurate results.

Output

Nodal output variable POR (pressure magnitude) is available for an acoustic medium in Abaqus (in Abaqus/CAE this output variable is called PAC). When the scattered wave formulation is used with incident wave loading in Abaqus/Explicit, output variable POR represents only the scattered pressure response of the model and does not include the incident wave loading itself. When the total wave formulation is used, output variable POR represents the total dynamic acoustic pressure, which includes contributions from both incident and scattered waves as well as the dynamic effects of fluid cavitation. For either formulation output variable POR does not include the acoustic static pressure, which is used only to evaluate the cavitation status in the acoustic medium.

In addition, in Abaqus/Standard nodal output variable PPOR (the pressure phase) is available for an acoustic medium. In Abaqus/Explicit nodal output variable PABS (the absolute pressure, equal to the sum of POR and the acoustic static pressure) is available for an acoustic medium.

Additional references

- Allard, J. F., M. Henry, J. Tizianel, L. Kelders, and W. Lauriks, "Sound Propagation in Air-Saturated Random Packings of Beads," Journal of the Acoustical Society of America, vol. 104, no. 4, p. 2004, 1998.

- Attenborough, K. F., “Acoustical Characterisitics of Rigid Fibrous Absorbents and Granular Materials,” Journal of the Acoustical Society of America, vol. 73, no. 3, p. 785, 1982.
- Craggs, A., “A Finite Element Model for Rigid Porous Absorbing Materials,” Journal of Sound and Vibration, vol. 61, no. 1, p. 101, 1978.
- Craggs, A., “Coupling of Finite Element Acoustic Absorption Models,” Journal of Sound and Vibration, vol. 66, no. 4, p. 605, 1979.
- Delany, M. E., and E. N. Bazley, “Acoustic Properties of Fibrous Absorbent Materials,” Applied Acoustics, vol. 3, p. 105, 1970.
- Miki, Y., “Acoustical Properties of Porous Materials - Modifications of Delany-Bazley Models,” Journal of the Acoustical Society of Japan (E), vol. 11, no. 1, p. 19, 1990.
- Song, B. H., and J. S. Bolton, “A Transfer-Matrix Approach for Estimating the Characteristic Impedance and Wavenumbers of Limp and Rigid Porous Materials,” Journal of the Acoustical Society of America, vol. 107, no. 3, p. 1131, 1999.
- Wilson, D. K., “Relaxation-Matched Modeling of Propagation through Porous Media, Including Fractal Pore Structure,” Journal of the Acoustical Society of America, vol. 94, no. 2, p. 1136, 1993.

26.4 Mass diffusion properties

- "Diffusivity," Section 26.4.1
- "Solubility," Section 26.4.2

26.4.1 DIFFUSIVITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Mass diffusion analysis," Section 6.9.1
- "Material library: overview," Section 21.1.1
- *DIFFUSIVITY
- *KAPPA
- "Defining mass diffusion," Section 12.12.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Diffusivity:

- defines the diffusion or movement of one material through another, such as the diffusion of hydrogen through a metal;
- must always be defined for mass diffusion analysis;
- must be defined in conjunction with "Solubility," Section 26.4.2;
- can be defined as a function of concentration, temperature, and/or predefined field variables;
- can be used in conjunction with a "Soret effect" factor to introduce mass diffusion caused by temperature gradients;
- can be used in conjunction with a pressure stress factor to introduce mass diffusion caused by gradients of equivalent pressure stress (hydrostatic pressure); and
- can produce a nonlinear mass diffusion analysis when dependence on concentration is included (the same can be said for the Soret effect factor and the pressure stress factor).

Defining diffusivity

Diffusivity is the relationship between the concentration flux, $\mathbf{J}$, of the diffusing material and the gradient of the chemical potential that is assumed to drive the mass diffusion process. Either general mass diffusion behavior or Fick's diffusion law can be used to define diffusivity, as discussed below.

General chemical potential

Diffusive behavior provides the following general chemical potential:

$$\mathbf{J} = -s\mathbf{D} \cdot \left[\frac{\partial \phi}{\partial \mathbf{x}} + \kappa_s \frac{\partial}{\partial \mathbf{x}} \left(\ln(\theta - \theta^Z) \right) + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right],$$

where

$\mathbf{D}(c, \theta, f_i)$	is the diffusivity;
$s(\theta, f_i)$	is the solubility (see "Solubility," Section 26.4.2);
$\kappa_s(c, \theta, f_i)$	is the Soret effect factor, providing diffusion because of temperature gradient (see below);
$\kappa_p(c, \theta, f_i)$	is the pressure stress factor, providing diffusion because of the gradient of the equivalent pressure stress (see below);
$\phi \stackrel{\text{def}}{=} c/s$	is the normalized concentration;
c	is the concentration of the diffusing material;
θ	is the temperature;
θ^Z	is the temperature at absolute zero (see below);
$p \stackrel{\text{def}}{=} -\text{trace}(\boldsymbol{\sigma})/3$	is the equivalent pressure stress; and
f_i	are any predefined field variables.

Input File Usage: *DIFFUSIVITY, LAW=GENERAL (default)

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: General**

Fick's law

An extended form of Fick's law can be used as an alternative to the general chemical potential:

$$\mathbf{J} = -\mathbf{D} \cdot \left(\frac{\partial c}{\partial \mathbf{x}} + s\kappa_p \frac{\partial p}{\partial \mathbf{x}} \right).$$

Input File Usage: *DIFFUSIVITY, LAW=FICK

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: Fick**

Directional dependence of diffusivity

Isotropic, orthotropic, or fully anisotropic diffusivity can be defined. For non-isotropic diffusivity a local orientation of the material directions must be specified (see "Orientations," Section 2.2.5).

Isotropic diffusivity

For isotropic diffusivity only one value of diffusivity is needed at each concentration, temperature, and field variable value.

Input File Usage: *DIFFUSIVITY, TYPE=ISO

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Type: Isotropic**

Orthotropic diffusivity

For orthotropic diffusivity three values of diffusivity (D_{11}, D_{22}, D_{33}) are needed at each concentration, temperature, and field variable value.

Input File Usage: *DIFFUSIVITY, TYPE=ORTHO

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Type: Orthotropic**

Anisotropic diffusivity

For fully anisotropic diffusivity six values of diffusivity (D_{11}, D_{12}, D_{22}, D_{13}, D_{23}, D_{33}) are needed at each concentration, temperature, and field variable value.

Input File Usage: *DIFFUSIVITY, TYPE=ANISO

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Type: Anisotropic**

Temperature-driven mass diffusion

The Soret effect factor, κ_s, governs temperature-driven mass diffusion. It can be defined as a function of concentration, temperature, and/or field variables in the context of the constitutive equation presented above. The Soret effect factor cannot be specified in conjunction with Fick's law since it is calculated automatically in this case (see "Mass diffusion analysis," Section 6.9.1).

Input File Usage: Use both of the following options to specify general temperature-driven mass diffusion:

*DIFFUSIVITY, LAW=GENERAL
*KAPPA, TYPE=TEMP

Use the following option to specify temperature-driven diffusion governed by Fick's law:

*DIFFUSIVITY, LAW=FICK

Abaqus/CAE Usage: Use the following options to specify general temperature-driven mass diffusion:

Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: General**: **Suboptions→Soret Effect**

Use the following option to specify temperature-driven diffusion governed by Fick's law:

Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: Fick**

Pressure stress-driven mass diffusion

The pressure stress factor, κ_p, governs mass diffusion driven by the gradient of the equivalent pressure stress. It can be defined as a function of concentration, temperature, and/or field variables in the context of the constitutive equation presented above.

Input File Usage: Use both of the following options:

*DIFFUSIVITY, LAW=GENERAL
*KAPPA, TYPE=PRESS

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: General**: **Suboptions→Pressure Effect**

Mass diffusion driven by both temperature and pressure stress

Specifying both κ_s and κ_p causes gradients of temperature and equivalent pressure stress to drive mass diffusion.

Input File Usage: Use all of the following options to specify general diffusion driven by gradients of temperature and pressure stress:

*DIFFUSIVITY, LAW=GENERAL
*KAPPA, TYPE=TEMP
*KAPPA, TYPE=PRESS

Use both of the following options to specify diffusion driven by the extended form of Fick's law:

*DIFFUSIVITY, LAW=FICK
*KAPPA, TYPE=PRESS

Abaqus/CAE Usage: Use the following options to specify general diffusion driven by gradients of temperature and pressure stress:

Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: General**: **Suboptions→Soret Effect** and **Suboptions→Pressure Effect**

Use the following options to specify diffusion driven by the extended form of Fick's law:

Property module: material editor: **Other→Mass Diffusion→Diffusivity**: **Law: Fick**: **Suboptions→Pressure Effect**

Specifying the value of absolute zero

You can specify the value of absolute zero as a physical constant.

Input File Usage: *PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Any module: **Model→Edit Attributes→*model_name*: Absolute zero temperature**

Elements

The mass diffusion law can be used only with the two-dimensional, three-dimensional, and axisymmetric solid elements that are included in the heat transfer/mass diffusion element library.

26.4.2 SOLUBILITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Mass diffusion analysis," Section 6.9.1
- "Material library: overview," Section 21.1.1
- *SOLUBILITY
- "Defining solubility" in "Defining mass diffusion," Section 12.12.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Solubility:

- is needed only for mass diffusion analysis;
- is also known as Sievert's parameter (in Sievert's law);
- must always accompany a diffusivity definition (see "Diffusivity," Section 26.4.1); and
- can be defined as a function of temperature and/or predefined field variables.

Defining solubility

Solubility, s, is used to define the "normalized concentration," ϕ, of the diffusing phase in a mass diffusion process:

$$\phi = c/s,$$

where c is the concentration. The normalized concentration is often also referred to as the "activity" of the diffusing material, and the gradients of the normalized concentration, along with gradients of temperature and pressure stress, drive the diffusion process (see "Diffusivity," Section 26.4.1).

Input File Usage: *SOLUBILITY

Abaqus/CAE Usage: Property module: material editor: **Other→Mass Diffusion→Solubility**

Elements

The mass diffusion law can be used only with the two-dimensional, three-dimensional, and axisymmetric solid elements that are included in the heat transfer/mass diffusion element library.

26.5 Electromagnetic properties

- "Electrical conductivity," Section 26.5.1
- "Piezoelectric behavior," Section 26.5.2
- "Magnetic permeability," Section 26.5.3

26.5.1 ELECTRICAL CONDUCTIVITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- *ELECTRICAL CONDUCTIVITY
- "Defining electrical conductivity," Section 12.11.1 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A material's electrical conductivity:

- must be defined for "Coupled thermal-electrical analysis," Section 6.7.3;
- must be defined for "Fully coupled thermal-electrical-structural analysis," Section 6.7.4;
- must be used to define the electromagnetic response of a conductor for "Eddy current analysis," Section 6.7.5;
- can be linear or nonlinear (by defining it as a function of temperature);
- can be isotropic, orthotropic, or fully anisotropic;
- can be specified as a function of temperature and/or field variables; and
- can be specified as a function of frequency for "Eddy current analysis," Section 6.7.5.

Directional dependence of electrical conductivity

Isotropic, orthotropic, or fully anisotropic electrical conductivity can be defined. For non-isotropic conductivity a local orientation for the material directions must be specified ("Orientations," Section 2.2.5).

Isotropic electrical conductivity

For isotropic electrical conductivity only one value of electrical conductivity is needed at each temperature and field variable value. Isotropic electrical conductivity is the default.

Input File Usage: *ELECTRICAL CONDUCTIVITY, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Electrical Conductivity**: **Type: Isotropic**

Orthotropic electrical conductivity

For orthotropic electrical conductivity three values of electrical conductivity (σ_{11}^E, σ_{22}^E, σ_{33}^E) are needed at each temperature and field variable value.

Input File Usage: *ELECTRICAL CONDUCTIVITY, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Electrical Conductivity**: **Type: Orthotropic**

Anisotropic electrical conductivity

For fully anisotropic electrical conductivity six values (σ^E_{11}, σ^E_{12}, σ^E_{22}, σ^E_{13}, σ^E_{23}, σ^E_{33}) are needed at each temperature and field variable value.

Input File Usage: *ELECTRICAL CONDUCTIVITY, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Electrical Conductivity**: **Type: Anisotropic**

Frequency-dependent electrical conductivity

Electrical conductivity can be defined as a function of frequency in an eddy current analysis.

Input File Usage: *ELECTRICAL CONDUCTIVITY, FREQUENCY

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Electrical Conductivity**: Toggle on **Use frequency-dependent data**

Elements

Electrical conductivity is active only in coupled thermal-electrical elements, coupled thermal-electrical-structural elements, and electromagnetic elements (see "Choosing the appropriate element for an analysis type," Section 27.1.3).

26.5.2 PIEZOELECTRIC BEHAVIOR

Products: Abaqus/Standard Abaqus/CAE

References

- "Piezoelectric analysis," Section 6.7.2
- "Material library: overview," Section 21.1.1
- *DIELECTRIC
- *PIEZOELECTRIC
- "Defining dielectric material properties," Section 12.11.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual
- "Defining piezoelectric properties," Section 12.11.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A piezoelectric material:

- is one in which an electrical field causes the material to strain, while stress causes an electric potential gradient;
- provides linear relations between mechanical and electrical fields; and
- is used in piezoelectric elements, which have both displacement and electrical potential as nodal variables.

Defining a piezoelectric material

A piezoelectric material responds to an electric potential gradient by straining, while stress causes an electric potential gradient in the material. This coupling between electric potential gradient and strain is the material's piezoelectric property. The material will also have a dielectric property so that an electrical charge exists when the material has a potential gradient. Piezoelectric material behavior is discussed in "Piezoelectric analysis," Section 2.10.1 of the Abaqus Theory Manual.

The mechanical properties of the material must be modeled by linear elasticity ("Linear elastic behavior," Section 22.2.1). The mechanical behavior can be defined by

$$\sigma_{ij} = D^E_{ijkl}\varepsilon_{kl} - e^{\varphi}_{mij}E_m,$$

in terms of the piezoelectric stress coefficient matrix, e^{φ}_{mij}, or by

$$\sigma_{ij} = D^E_{ijkl}(\varepsilon_{kl} - d^{\varphi}_{mkl}E_m),$$

in terms of the piezoelectric strain coefficient matrix, d^{φ}_{mkl}. The electrical behavior is defined by

$$q_i = e^{\varphi}_{ijk}\varepsilon_{jk} + D^{\varphi(\varepsilon)}_{ij}E_j,$$

where

σ_{ij} is the mechanical stress tensor;

ε_{ij} is the strain tensor;

q_i is the electric "displacement" vector;

D^{E}_{ijkl} is the material's elastic stiffness matrix defined at zero electrical potential gradient (short circuit condition);

e^{φ}_{mij} is the material's piezoelectric stress coefficient matrix, defining the stress σ_{ij} caused by the electrical potential gradient E_m in a fully constrained material (it can also be interpreted as the electrical displacement q_m caused by the applied strain ε_{ij} at a zero electrical potential gradient);

d^{φ}_{mkl} is the material's piezoelectric strain coefficient matrix, defining the strain ε_{kl} caused by the electrical potential gradient E_m in an unconstrained material (an alternative interpretation is given later in this section);

φ is the electrical potential;

$D^{\varphi(\varepsilon)}_{ij}$ is the material's dielectric property, defining the relation between the electric displacement q_i and the electric potential gradient E_j for a fully constrained material; and

E_i is the electrical potential gradient vector, $-\partial\varphi/\partial x_i$.

The material's electrical and electro-mechanical coupling behaviors are, thus, defined by its dielectric property, $D^{\varphi(\varepsilon)}_{ij}$, and its piezoelectric stress coefficient matrix, e^{φ}_{mij}, or its piezoelectric strain coefficient matrix, d^{φ}_{mkl}. These properties are defined as part of the material definition ("Material data definition," Section 21.1.2).

Alternative forms of the constitutive equations

Alternative forms of the piezoelectric constitutive equations are presented in this section. These forms of the equations involve material properties that cannot be used directly as input for Abaqus/Standard. However, they are related to the Abaqus/Standard input through simple relations that are presented in "Piezoelectric analysis," Section 2.10.1 of the Abaqus Theory Manual. The intent of this section is to draw connections between the Abaqus/Standard terminology and input to that used commonly in the piezoelectricity community. The mechanical behavior can also be defined by

$$\sigma_{ij} = D^{q}_{ijkl}(\varepsilon_{kl} - g^{\varphi}_{mkl}q_m)$$

in terms of the piezoelectric coefficient matrix g^{φ}_{mij}, and the stiffness matrix D^{q}_{ijkl}, which defines the mechanical properties at zero electrical displacement (open circuit condition). Likewise, the electrical behavior can also be defined by

$$q_i = d^{\varphi}_{ijk}\sigma_{jk} + D^{\varphi(\sigma)}_{ij}E_j$$

in terms of the dielectric matrix $D_{ij}^{\varphi(\sigma)}$ for an unconstrained material or by

$$q_i = D_{im}^{\varphi(\sigma)} g_{mjk}^{\varphi} \sigma_{jk} + D_{ij}^{\varphi(\sigma)} E_j,$$

where

D_{ijkl}^{q} is the material's elastic stiffness matrix defined at zero electrical displacement;

d_{ijk}^{φ} is the material's piezoelectric strain coefficient matrix used earlier, and based on the equations, may alternatively be interpreted as the electrical displacement q_i caused by the stress σ_{jk} at zero electrical potential gradient;

g_{mkl}^{φ} is the material's piezoelectric coefficient matrix, which can be interpreted as defining either the strain ε_{kl} caused by the electrical displacement q_m in an unconstrained material or the electrical potential gradient E_m caused by the stress σ_{kl} at zero electrical displacement; and

$D_{ij}^{\varphi(\sigma)}$ is the material's dielectric property, defining the relation between the electric displacement q_i and the electric potential gradient E_j for an unconstrained material.

These are useful relationships that are often seen in the piezoelectric literature. In "Piezoelectric analysis," Section 2.10.1 of the Abaqus Theory Manual, the properties g_{mkl}^{φ}, $D_{ij}^{\varphi(\sigma)}$, and D_{ijkl}^{q} are expressed in terms of the properties d_{mkl}^{φ}, $D_{ij}^{\varphi(\varepsilon)}$, and D_{ijkl}^{E}, that are used as input for Abaqus/Standard.

Specifying dielectric material properties

The dielectric matrix can be isotropic, orthotropic, or fully anisotropic. For non-isotropic dielectric materials a local orientation for the material directions must be specified ("Orientations," Section 2.2.5). The entries of the dielectric matrix (what are referred to as "dielectric constants" in Abaqus) refer to what is more commonly known in the literature as the permittivity of the material.

Isotropic dielectric properties

The dielectric matrix $D_{ij}^{\varphi(\varepsilon)}$ can be fully isotropic, so that

$$D_{ij}^{\varphi(\varepsilon)} = D^{\varphi(\varepsilon)} \delta_{ij}.$$

You specify the single value $D^{\varphi(\varepsilon)}$ for the dielectric constant. $D^{\varphi(\varepsilon)}$ must be determined for a constrained material. Isotropic behavior is the default.

Input File Usage: *DIELECTRIC, TYPE=ISO

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Dielectric (Electrical Permittivity)**: **Type: Isotropic**

Orthotropic dielectric properties

For orthotropic behavior you must specify three values in the dielectric matrix ($D_{11}^{\varphi(\varepsilon)}$, $D_{22}^{\varphi(\varepsilon)}$, and $D_{33}^{\varphi(\varepsilon)}$).

Input File Usage: *DIELECTRIC, TYPE=ORTHO

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Dielectric (Electrical Permittivity)**: **Type: Orthotropic**

Anisotropic dielectric properties

For fully anisotropic behavior you must specify six values in the dielectric matrix ($D_{11}^{\varphi(\varepsilon)}$, $D_{12}^{\varphi(\varepsilon)}$, $D_{22}^{\varphi(\varepsilon)}$, $D_{13}^{\varphi(\varepsilon)}$, $D_{23}^{\varphi(\varepsilon)}$, and $D_{33}^{\varphi(\varepsilon)}$).

Input File Usage: *DIELECTRIC, TYPE=ANISO

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Dielectric (Electrical Permittivity)**: **Type: Anisotropic**

Specifying piezoelectric material properties

The piezoelectric material properties can be defined by giving the stress coefficients, e_{mij}^{φ} (this is the default), or by giving the strain coefficients, d_{mkl}^{φ}. In either case, 18 components must be given in the following order (substitute $\boldsymbol{d}$ for $\boldsymbol{e}$ for strain coefficients):

$$\begin{array}{llllll} e_{1\,11}^{\varphi}, & e_{1\,22}^{\varphi}, & e_{1\,33}^{\varphi}, & e_{1\,12}^{\varphi}, & e_{1\,13}^{\varphi}, & e_{1\,23}^{\varphi}, \\ e_{2\,11}^{\varphi}, & e_{2\,22}^{\varphi}, & e_{2\,33}^{\varphi}, & e_{2\,12}^{\varphi}, & e_{2\,13}^{\varphi}, & e_{2\,23}^{\varphi}, \\ e_{3\,11}^{\varphi}, & e_{3\,22}^{\varphi}, & e_{3\,33}^{\varphi}, & e_{3\,12}^{\varphi}, & e_{3\,13}^{\varphi}, & e_{3\,23}^{\varphi}. \end{array}$$

The first index on these coefficients refers to the component of electric displacement (sometimes called the electric flux), while the last pair of indices refers to the component of mechanical stress or strain.

Thus, the piezoelectric components causing electrical displacement in the 1-direction are all given first, then those causing electrical displacement in the 2-direction, and then those causing electrical displacement in the 3-direction. (Some references list these coupling terms in a different order.)

Input File Usage: Use the following option to give the stress coefficients:

*PIEZOELECTRIC, TYPE=S

Use the following option to give the strain coefficients:

*PIEZOELECTRIC, TYPE=E

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Piezoelectric**: **Type: Stress** or **Strain**

Converting double index notation to triple index notation

Industry-supplied piezoelectric data often use a double index notation. A double index notation can be converted easily to the required triple index notation in Abaqus/Standard by noting the convention followed in Abaqus for the correspondence between (second-order) tensor and vector notations: the 11, 22, 33, 12, 13, and 23 components of the tensor correspond to the 1, 2, 3, 4, 5, and 6 components, respectively, of the corresponding vector.

Elements

Piezoelectric coupling is active only in piezoelectric elements (those with displacement degrees of freedom and electrical potential degree of freedom 9). See “Choosing the appropriate element for an analysis type,” Section 27.1.3.

26.5.3 MAGNETIC PERMEABILITY

Products: Abaqus/Standard Abaqus/CAE

References

- "Material library: overview," Section 21.1.1
- *MAGNETIC PERMEABILITY
- *NONLINEAR BH
- "Defining magnetic permeability," Section 12.11.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

A material's magnetic permeability:

- must be defined for "Eddy current analysis," Section 6.7.5, and "Magnetostatic analysis," Section 6.7.6;
- can be specified directly for linear magnetic behavior or through one or more B–H curves for nonlinear magnetic behavior;
- can be isotropic, orthotropic, or (in the case of linear behavior) fully anisotropic;
- can be specified as a function of temperature and/or field variables; and
- can be specified as a function of frequency in a time-harmonic eddy current procedure.

Linear magnetic behavior

Linear magnetic behavior is defined by direct specification of magnetic permeability.

Directional dependence of magnetic permeability

Isotropic, orthotropic, or fully anisotropic magnetic permeability can be defined. For non-isotropic magnetic permeability a local orientation for the material directions must be specified ("Orientations," Section 2.2.5).

Isotropic magnetic permeability

For isotropic magnetic permeability only one value of magnetic permeability is needed at each temperature and field variable value. Isotropic magnetic permeability is the default.

Input File Usage: *MAGNETIC PERMEABILITY, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Magnetic Permeability**: **Type: Isotropic**

Orthotropic magnetic permeability

For orthotropic magnetic permeability three values of magnetic permeability (μ_{11}, μ_{22}, μ_{33}) are needed at each temperature and field variable value.

Input File Usage: *MAGNETIC PERMEABILITY, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Magnetic Permeability**: **Type: Orthotropic**

Anisotropic magnetic permeability

For fully anisotropic magnetic permeability six values (μ_{11}, μ_{12}, μ_{22}, μ_{13}, μ_{23}, μ_{33}) are needed at each temperature and field variable value.

Input File Usage: *MAGNETIC PERMEABILITY, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Magnetic Permeability**: **Type: Anisotropic**

Frequency-dependent magnetic permeability

Magnetic permeability can be defined as a function of frequency in a time-harmonic eddy current analysis.

Input File Usage: *MAGNETIC PERMEABILITY, FREQUENCY

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic→Magnetic Permeability**: Toggle on **Use frequency-dependent data**

Nonlinear magnetic behavior

Nonlinear magnetic behavior is characterized by magnetic permeability that depends on the strength of the magnetic field. The nonlinear magnetic material model in Abaqus is suitable for ideally soft magnetic materials (without any hysteresis effects) characterized by a monotonically increasing response in B–H space, where B and H refer to the strengths of the magnetic flux density vector and the magnetic field vector, respectively. Nonlinear magnetic behavior is defined through direct specification of one or more B–H curves that provide B as a function of H and, optionally, temperature and/or predefined field variables, in one or more directions. Nonlinear magnetic behavior can be isotropic, orthotropic, or transversely isotropic (which is a special case of the more general orthotropic behavior). More than one B–H curve is needed to define the nonlinear magnetic behavior if it is not isotropic.

Directional dependence of nonlinear magnetic behavior

Isotropic, orthotropic, or transversely isotropic nonlinear magnetic behavior can be defined. For non-isotropic nonlinear magnetic behavior a local orientation for the material directions must be specified (“Orientations,” Section 2.2.5).

Isotropic nonlinear magnetic behavior

For isotropic nonlinear magnetic response only one B–H curve is needed at each temperature and field variable value. Isotropic magnetic permeability is the default. Abaqus assumes that the nonlinear magnetic behavior is governed by

$$\mathbf{B} = B\left(|\mathbf{H}|\right)\left(\frac{\mathbf{H}}{|\mathbf{H}|}\right)$$

Input File Usage: You define $B\left(|\mathbf{H}|\right)$ through a B–H curve:

∗MAGNETIC PERMEABILITY, NONLINEAR, TYPE=ISOTROPIC
∗NONLINEAR BH, DIR=*direction*

The B–H curve in any direction (i.e., the nonlinear behavior in global direction 1, 2, or 3) will suffice as the nonlinear magnetic behavior is assumed to be the same in all directions.

Abaqus/CAE Usage: Nonlinear magnetic behavior is not supported in Abaqus/CAE.

Orthotropic nonlinear magnetic behavior

For orthotropic nonlinear magnetic response three B–H curves (one curve to define the behavior in each of the local directions 1, 2, and 3) are needed at each temperature and field variable value. Abaqus assumes that the nonlinear magnetic behavior in the local material directions is governed by

$$\mathbf{B} = diag\left(B_1\left(|\mathbf{H}|\right), B_2\left(|\mathbf{H}|, B_3\left(|\mathbf{H}|\right)\right)\right)\left(\frac{\mathbf{H}}{|\mathbf{H}|}\right),$$

where $diag\left(\right)$ refers to a diagonal matrix.

Transversely isotropic nonlinear magnetic behavior is a special case of orthotropic behavior, in which the behavior in any two directions is the same and is different from that in the third direction.

Input File Usage: You define $B_1\left(|\mathbf{H}|\right)$, $B_2\left(|\mathbf{H}|\right)$, and $B_3\left(|\mathbf{H}|\right)$, respectively, through three independent B–H curves, one in each of the directions 1, 2, and 3:

∗MAGNETIC PERMEABILITY, NONLINEAR, TYPE=ORTHOTROPIC
∗NONLINEAR BH, DIR=1
…
∗NONLINEAR BH, DIR=2
…
∗NONLINEAR BH, DIR=3
…

Abaqus/CAE Usage: Nonlinear magnetic behavior is not supported in Abaqus/CAE.

Elements

Magnetic material behavior is active only in electromagnetic elements (see “Choosing the appropriate element for an analysis type,” Section 27.1.3).

26.6 Pore fluid flow properties

- "Pore fluid flow properties," Section 26.6.1
- "Permeability," Section 26.6.2
- "Porous bulk moduli," Section 26.6.3
- "Sorption," Section 26.6.4
- "Swelling gel," Section 26.6.5
- "Moisture swelling," Section 26.6.6

26.6.1 PORE FLUID FLOW PROPERTIES

Abaqus/Standard allows specific properties to be defined for a fluid-filled porous material. This type of porous medium is considered in a coupled pore fluid diffusion/stress analysis ("Coupled pore fluid diffusion and stress analysis," Section 6.8.1). The following properties are available:

- **Permeability:** Permeability defines the relationship between the flow rate of a liquid through a porous medium and the gradient of the piezometric head of that fluid (see "Permeability," Section 26.6.2).
- **Porous bulk moduli:** The bulk moduli of the solid grains and of the fluid in a porous medium are defined such that their compressibility is considered in an analysis (see "Porous bulk moduli," Section 26.6.3).
- **Sorption:** Sorption defines the absorption/exsorption behavior of a porous material under partially saturated flow conditions (see "Sorption," Section 26.6.4).
- **Swelling gel:** The swelling gel model is used to simulate the growth of gel particles that swell and trap wetting liquid in a partially saturated porous medium (see "Swelling gel," Section 26.6.5).
- **Moisture swelling:** Moisture swelling defines the saturation-driven volumetric swelling of a porous medium's solid skeleton under partially saturated flow conditions (see "Moisture swelling," Section 26.6.6).

Thermal expansion

For porous media such as soils or rock, the thermal expansion of both the solid grains and the permeating fluid can be defined. See "Thermal expansion" in "Coupled pore fluid diffusion and stress analysis," Section 6.8.1, for more details.

26.6.2 PERMEABILITY

Products: Abaqus/Standard Abaqus/CFD Abaqus/CAE

References

- "Pore fluid flow properties," Section 26.6.1
- "Material library: overview," Section 21.1.1
- *PERMEABILITY
- "Defining permeability" in "Defining a fluid-filled porous material," Section 12.12.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Permeability is the relationship between the volumetric flow rate per unit area of a particular wetting liquid through a porous medium and the gradient of the effective fluid pressure. It can be specified in Abaqus/Standard and Abaqus/CFD.

Permeability in Abaqus/Standard:

- must be specified for a wetting liquid for an effective stress/wetting liquid diffusion analysis (see "Coupled pore fluid diffusion and stress analysis," Section 6.8.1);
- is defined, in general, by Forchheimer's law, which accounts for changes in permeability as a function of fluid flow velocity; and
- can be isotropic, orthotropic, or fully anisotropic and can be given as a function of void ratio, saturation, temperature, and field variables.

Permeability in Abaqus/CFD:

- must be specified for porous media flows (see "Incompressible fluid dynamic analysis," Section 6.6.2); and
- can be isotropic and specified as a function of porosity only or can be specified through the Carman-Kozeny permeability-porosity relation.

Permeability in Abaqus/Standard

Permeability is defined for pore fluid flow.

Forchheimer's law

According to Forchheimer's law, high flow velocities have the effect of reducing the effective permeability and, therefore, "choking" pore fluid flow. As the fluid flow velocity reduces, Forchheimer's law approximates the well-known Darcy's law. Darcy's law can, therefore, be used directly in Abaqus/Standard by omitting the velocity-dependent term in Forchheimer's law.

Forchheimer's law is written as

$$\mathbf{f}(1+\beta\sqrt{\mathbf{v}_w\cdot\mathbf{v}_w}) = -\frac{k_s}{\gamma_w}\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w\mathbf{g}\right),$$

where

$\mathbf{f} = sn\mathbf{v}_w$	is the volumetric flow rate of wetting liquid per unit area of the porous medium (the effective velocity of the wetting liquid);
$s = \frac{dV_w}{dV_v}$	is the fluid saturation ($s = 1$ for a fully saturated medium, $s = 0$ for a completely dry medium);
$n = \frac{dV_v}{dV}$	is the porosity of the porous medium;
$e = \frac{dV_v}{(dV_g+dV_t)}$	is the void ratio;
dV_w	is the wetting fluid volume in the medium;
dV_v	is the void volume in the medium;
dV_g	is the volume of grains of solid material in the medium;
dV_t	is the volume of trapped wetting liquid in the medium;
dV	is the total volume of the medium;
$\mathbf{v}_w$	is the fluid velocity;
$\beta(e)$	is a "velocity coefficient," which may be dependent on the void ratio of the material;
$k_s(s)$	is the dependence of permeability on saturation of the wetting liquid such that $k_s = 1.0$ at $s = 1.0$;
$\rho_w = \gamma_w/g$	is the density of the fluid;
γ_w	is the specific weight of the wetting liquid;
g	is the magnitude of the gravitational acceleration;
$\mathbf{k}(e,\theta,f_\beta)$	is the permeability of the fully saturated medium, which can be a function of void ratio (e, common in soil consolidation problems), temperature (θ), and/or field variables (f_β);
u_w	is the wetting liquid pore pressure;
$\mathbf{x}$	is position; and
$\mathbf{g}$	is the gravitational acceleration.

Permeability definitions

Permeability can be defined in different ways by different authors; caution should, therefore, be used to ensure that the specified input data are consistent with the definitions used in Abaqus/Standard.

Permeability in Abaqus/Standard is defined as

$$\overline{\mathbf{k}} = \frac{k_s}{(1+\beta\sqrt{\mathbf{v}_w\cdot\mathbf{v}_w})}\mathbf{k},$$

so that Forchheimer's law can also be written as

$$\mathbf{f} = -\frac{\overline{\mathbf{k}}}{\gamma_w} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g} \right).$$

The fully saturated permeability, $\mathbf{k}$, is typically obtained from experiments under low fluid velocity conditions. $\mathbf{k}$ can be defined as a function of void ratio, e, (common in soil consolidation problems) and/or temperature, θ. The void ratio can be derived from the porosity, n, using the relationship $e = n/(1-n)$. Up to six variables may be needed to define the fully saturated permeability, depending on whether isotropic, orthotropic, or fully anisotropic permeability is to be modeled (discussed below).

Alternative definition of permeability

Some authors refer to the definition of permeability used in Abaqus/Standard, $\overline{\mathbf{k}}$ (units of LT^{-1}), as the "hydraulic conductivity" of the porous medium and define the permeability as

$$\widehat{\mathbf{K}} = \frac{\nu}{g} \frac{k_s}{(1 + \beta \sqrt{\mathbf{v}_w \cdot \mathbf{v}_w})} \mathbf{k} = \frac{\nu}{g} \overline{\mathbf{k}},$$

where ν is the kinematic viscosity of the wetting liquid (the ratio of the liquid's dynamic viscosity to its mass density), g is the magnitude of the gravitational acceleration, and $\widehat{\mathbf{K}}$ has dimensions L^2 (or Darcy). If the permeability is available in this form, it must be converted such that the appropriate values of $\mathbf{k}$ are used in Abaqus/Standard.

Specifying the permeability

Permeability in Abaqus/Standard can be isotropic, orthotropic, or fully anisotropic. For non-isotropic permeability a local orientation (see "Orientations," Section 2.2.5) must be used to specify the material directions.

Isotropic permeability

For isotropic permeability in Abaqus/Standard define one value of the fully saturated permeability at each value of the void ratio.

Input File Usage: *PERMEABILITY, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Type: Isotropic**

Orthotropic permeability

For orthotropic permeability in Abaqus/Standard define three values of the fully saturated permeability (k_{11}, k_{22}, and k_{33}) at each value of the void ratio.

Input File Usage: *PERMEABILITY, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Type: Orthotropic**

Anisotropic permeability

For fully anisotropic permeability in Abaqus/Standard define six values of the fully saturated permeability (k_{11}, k_{12}, k_{22}, k_{13}, k_{23}, and k_{33}) at each value of the void ratio.

Input File Usage: *PERMEABILITY, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Type: Anisotropic**

Velocity coefficient

Abaqus/Standard assumes that $\beta = 0.0$ by default, meaning that Darcy's law is used. If Forchheimer's law is required ($\beta > 0.0$), $\beta(e)$ must be defined in tabular form.

Input File Usage: *PERMEABILITY, TYPE=VELOCITY

This must be a repeated use of the *PERMEABILITY option for the same material, since $\mathbf{k}(e, \theta)$ must also be defined.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Suboptions→Velocity Dependence**

Saturation dependence

In Abaqus/Standard you can define the dependence of permeability, $\overline{\mathbf{k}}$, on saturation, s, by specifying k_s. Abaqus/Standard assumes by default that $k_s = s^3$ for $s < 1.0$; $k_s = 1.0$ for $s \geq 1.0$. The tabular definition of $k_s(s)$ must specify $k_s = 1.0$ for $s \geq 1.0$.

Input File Usage: *PERMEABILITY, TYPE=SATURATION

This must be a repeated use of the *PERMEABILITY option for the same material, since $\mathbf{k}(e, \theta)$ must also be defined.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Suboptions→Saturation Dependence**

Specific weight of the wetting liquid

In Abaqus/Standard the specific weight of the fluid, γ_w, must be specified correctly even if the analysis does not consider the weight of the wetting liquid (i.e., if excess pore fluid pressure is calculated).

Input File Usage: *PERMEABILITY, TYPE=*type*, SPECIFIC=γ_w

The SPECIFIC parameter must be defined in conjunction with the fully saturated *PERMEABILITY option for a given medium.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Specific weight of wetting liquid:** γ_w

Permeability in Abaqus/CFD

For flows in fluid-saturated porous medium, the momentum equation in its simplest form can be written as

$$\nabla p = -\frac{\mu}{K}\mathbf{v} - \frac{\rho c_F}{K^{\frac{1}{2}}}|\mathbf{v}|\mathbf{v},$$

where the first term on the right-hand side is the Darcy drag and the second term is the inertial drag (also called form drag or Forchheimer drag). In the above equation

p	is the intrinsic average of the pressure (average taken over the fluid-phase only);
$\mathbf{v}$	is the extrinsic or superficial velocity vector, where the average is taken over a representative volume incorporating both the solid (matrix) and the fluid phases;
ρ	is the density of the fluid;
μ	is the viscosity of the fluid;
K	is the permeability of the porous medium (units of L^2); and
c_F	is the dimensionless inertial or form drag coefficient and, in general, is a function of the porosity ϵ.

The inertial drag coefficient, c_F, is usually a function of the porosity ϵ. In Abaqus/CFD the Ergun's relation is used, which is given by

$$c_F = \frac{C}{\sqrt{\epsilon^3}},$$

where the constant C is set by default as $\frac{1.75}{\sqrt{150}} = 0.142887$.

A widely used model to specify the permeability K as a function of porosity is the Carman-Kozeny relation, which is given by

$$K = \frac{r_f^2}{4k_{kc}}\frac{\epsilon^3}{(1-\epsilon)^2},$$

where k_{kc} represents the Carman-Kozeny constant (parameter that is geometry dependent) and r_f represents the average radius of the porous particles/fibers.

Specifying the permeability

Permeability in Abaqus/CFD can be isotropic (with dependence only on porosity) or specified using a Carman-Kozeny relation.

Isotropic permeability

For isotropic permeability define one value of the fully saturated permeability at each value of the porosity.

Input File Usage: *PERMEABILITY, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Type: Isotropic (CFD)**

Carman-Kozeny model

For the Carman-Kozeny relation, you can define the permeability K by specifying k_{kc}, the Carman-Kozeny constant, and r_f, the average pore-particle/fiber radius.

Input File Usage: *PERMEABILITY, TYPE=CARMAN KOZENY

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Type: Carman-Kozeny**

Inertial drag coefficient

The value of the constant C in the expression for the inertial drag coefficient, c_F, can be set to any user-specified value. By default, the value of C is 0.142887.

Input File Usage: *PERMEABILITY, TYPE=*type*, INERTIAL DRAG COEFFICIENT=C

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Permeability**: **Inertial drag coefficient:** C

Elements

In Abaqus/Standard permeability can be used only in elements that allow for pore pressure (see "Choosing the appropriate element for an analysis type," Section 27.1.3). Permeability can be used with any fluid element in Abaqus/CFD.

26.6.3 POROUS BULK MODULI

Products: Abaqus/Standard Abaqus/CAE

References

- "Pore fluid flow properties," Section 26.6.1
- "Material library: overview," Section 21.1.1
- *POROUS BULK MODULI
- "Defining porous bulk moduli" in "Defining a fluid-filled porous material," Section 12.12.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The porous bulk moduli:

- must be defined whenever the compressibility of the solid grains or the compressibility of the permeating fluid is to be considered in the analysis of a porous medium; and
- must be defined when a swelling gel is modeled ("Moisture swelling," Section 26.6.6).

Defining porous bulk moduli

You can specify the bulk modulus of the solid grains and the bulk modulus of the fluid as functions of temperature. If either modulus is omitted or set to zero, that phase of the material is assumed to be fully incompressible.

Input File Usage: *POROUS BULK MODULI

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Porous Bulk Moduli**

Elements

The porous bulk moduli can be defined only for elements that allow for pore pressure (see "Choosing the appropriate element for an analysis type," Section 27.1.3).

26.6.4 SORPTION

Products: Abaqus/Standard Abaqus/CAE

References

- "Pore fluid flow properties," Section 26.6.1
- "Material library: overview," Section 21.1.1
- *SORPTION
- "Defining sorption" in "Defining a fluid-filled porous material," Section 12.12.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Sorption:

- defines a porous material's absorption/exsorption behavior under partially saturated flow conditions; and
- is used in the analysis of coupled wetting liquid flow and porous medium stress ("Coupled pore fluid diffusion and stress analysis," Section 6.8.1).

Sorption

A porous medium becomes partially saturated when the total pore liquid pressure, u_w, becomes negative (see "Effective stress principle for porous media," Section 2.8.1 of the Abaqus Theory Manual). Negative values of u_w represent capillary effects in the medium. For $u_w < 0$ it is known that the saturation lies within certain limits that depend on the value of the capillary pressure, $-u_w$ (see "Continuity statement for the wetting liquid phase in a porous medium," Section 2.8.4 of the Abaqus Theory Manual). Typical forms of these limits are shown in Figure 26.6.4–1. We write these limits as $s^a \le s \le s^e$, where $s^a(u_w)$ is the limit at which absorption will occur (so that $\dot{s} > 0$), and $s^e(u_w)$ is the limit at which exsorption will occur (so that $\dot{s} < 0$). The transition between absorption and exsorption and vice versa takes place along "scanning" curves (discussed below). These curves are approximated by the single straight line shown in Figure 26.6.4–1.

When partial saturation is included in the analysis of flow through a porous medium, the absorption behavior, the exsorption behavior, and the scanning behavior (between absorption and exsorption) should each be defined. Each of these behaviors is discussed below. If sorption is not defined at all, Abaqus/Standard assumes fully saturated flow ($s = 1.0$) for all values of u_w.

Strongly unsymmetric partially saturated flow coupled equations result from the definition of sorption. Therefore, Abaqus/Standard automatically uses its unsymmetric matrix storage and solution scheme (see "Defining an analysis," Section 6.1.2) if you request partially saturated analysis (i.e., if sorption is defined).

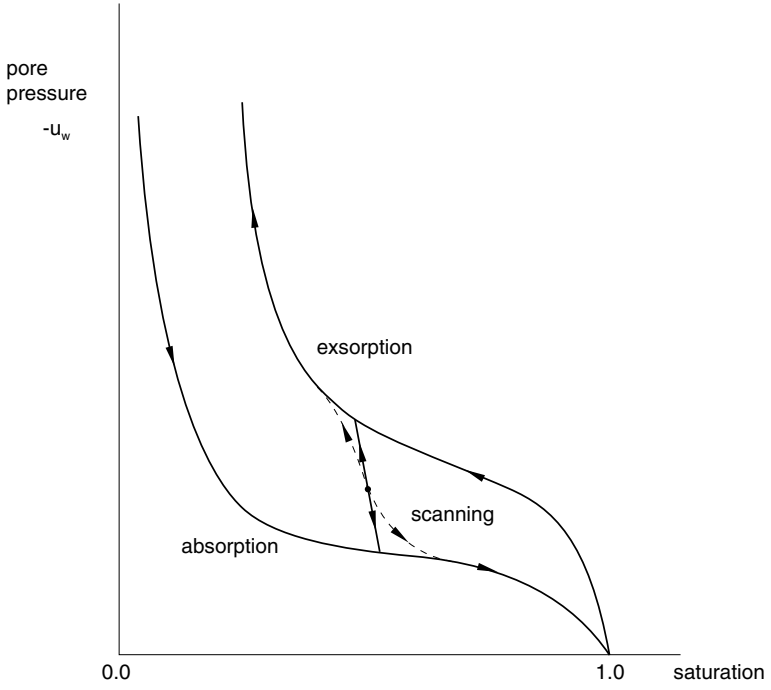


Figure 26.6.4–1 Typical absorption and exsorption behaviors.

Defining absorption and exsorption

Absorption and exsorption behaviors are defined by specifying the pore liquid pressure, u_w (negative "capillary tension"), as a function of saturation. In most physical cases the wetting liquid cannot be driven to zero saturation; to achieve zero saturation, the data would have to define $u_w \rightarrow -\infty$ as $s \rightarrow 0.0$. Absorption and exsorption data can be defined in either a tabular form or an analytical form.

Tabular form

By default, you define the absorption and exsorption behaviors by specifying u_w as a tabular function of s, where $0^+ \leq s \leq 1.0$.

Input File Usage: Use the following options:

∗SORPTION, TYPE=ABSORPTION, LAW=TABULAR
∗SORPTION, TYPE=EXSORPTION, LAW=TABULAR

If the ∗SORPTION option is used only once, the behavior defined is taken as the behavior for absorption and exsorption.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Sorption**
Absorption: **Law: Tabular**
Exsorption: toggle on **Include exsorption**: **Law: Tabular**

Analytical form

The absorption and exsorption behaviors can be defined by the following analytical form:

$$u_w = \frac{1}{B}\ln\left[\frac{(s - s_0)}{(1 - s_0) + A(1 - s)}\right] \quad \text{for} \quad s_1 \le s < 1,$$
$$u_w = u_w\big|_{s_1} - \left.\frac{du_w}{ds}\right|_{s_1} (s_1 - s) \qquad \text{for} \quad s_0 \le s < s_1,$$

where A, B are positive material constants and s_0, s_1 are parameters used to define the lower bound of the saturation values of interest (see Figure 26.6.4–2).

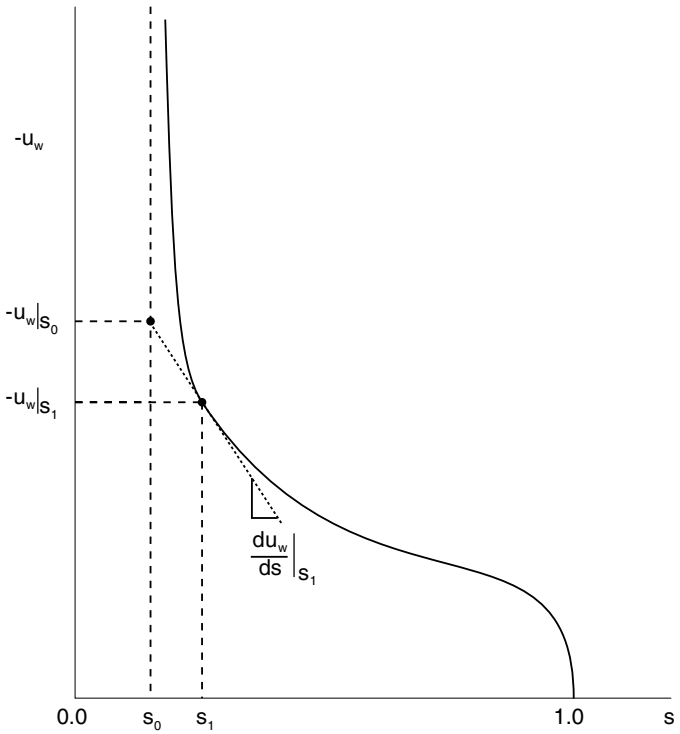


Figure 26.6.4–2 Logarithmic form of absorption and exsorption behaviors.

Input File Usage: Use the following options:

*SORPTION, TYPE=ABSORPTION, LAW=LOG
*SORPTION, TYPE=EXSORPTION, LAW=LOG

If the *SORPTION option is used only once, the behavior defined is taken as the behavior for absorption and exsorption.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Sorption**
Absorption: **Law: Log**
Exsorption: toggle on **Include exsorption**: **Law: Log**

Defining the behavior between absorption and exsorption

The behavior between absorption and exsorption is defined by a scanning line of user-specified constant slope, $(du_w/ds)\big|_s$. This slope should be larger than the slope of any segment of the absorption or exsorption behaviors.

If absorption and exsorption behaviors are defined with no scanning line, the slope of the scanning line is taken as 1.05 times the largest value of du_w/ds given in the absorption and exsorption behavior definitions.

Input File Usage: *SORPTION, TYPE=SCANNING

This must be a repeated use of the *SORPTION option for the same material.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Sorption**:
Exsorption: toggle on **Include exsorption** and **Include scanning**: **Slope** $(du_w/ds)\big|_s$

Elements

Sorption can be used only in elements that allow for pore pressure (see "Choosing the appropriate element for an analysis type," Section 27.1.3).

26.6.5 SWELLING GEL

Products: Abaqus/Standard Abaqus/CAE

References

- "Pore fluid flow properties," Section 26.6.1
- "Material library: overview," Section 21.1.1
- *GEL
- "Defining a swelling gel" in "Defining a fluid-filled porous material," Section 12.12.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

The swelling gel model:

- allows for modeling of the growth of gel particles that swell and trap wetting liquid in a partially saturated porous medium;
- is intended for use in moisture absorption problems, which typically involve polymeric materials, such as in the analysis of diapers; and
- can be used in the analysis of coupled pore liquid flow and porous medium stress (see "Coupled pore fluid diffusion and stress analysis," Section 6.8.1).

Swelling gel model

The simple swelling gel model is based on the idealization of a gel as a volume of individual spherical particles of equal radius, r_a. The swelling evolution (discussed in detail in "Constitutive behavior in a porous medium," Section 2.8.3 of the Abaqus Theory Manual) is assumed to be given by

$$\dot{r}_a = \frac{r_a^f - r_a}{\tau_1} \left\langle s - 1 + \left(\frac{(r_a^f)^3 - (r_a)^3}{(r_a^f)^3 - (r_a^{\mathrm{dry}})^3} \right) \right\rangle \left(1 - \left\langle \frac{r_a - r_a^t}{r_a^s - r_a^t} \right\rangle^2 \right),$$

where the value of any grouping of terms in angled brackets $\langle\ \rangle$ is set equal to zero if its mathematical result is not positive, and

r_a^f	is the fully swollen radius;
τ_1	is the relaxation time of the gel particles;
s	is the saturation of the surrounding medium;
r_a^{dry}	is the radius of the gel particles when they are completely dry;
$r_a^t = \left(\frac{n^0 J}{4\sqrt{2} k_a} \right)^{\frac{1}{3}}$	is the maximum radius that the gel particles can achieve before they must touch;
$r_a^s = \left(\frac{3}{4\pi} \frac{n^0 J}{k_a} \right)^{\frac{1}{3}}$	is the effective gel radius when the volume is entirely occupied with gel;

n^0	is the initial porosity of the material;
J	is the volume change in the material; and
k_a	is the number of gel particles per unit volume.

The second term in the definition of gel growth incorporates the assumption that the gel will swell only when the saturation of the surrounding medium, s, exceeds the effective saturation of the gel. The third term in the growth equation reduces the swelling rate when the surface of gel particles exposed to free fluid is limited by the combination of packing density and gel particle radius.

The swelling gel model is defined by specifying the variables r_a^{dry}, r_a^f, k_a, and τ_1.

Input File Usage: *GEL

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Gel**

Elements

The swelling gel model can be used only in elements that allow for pore pressure (see "Choosing the appropriate element for an analysis type," Section 27.1.3).

26.6.6 MOISTURE SWELLING

Products: Abaqus/Standard Abaqus/CAE

References

- "Pore fluid flow properties," Section 26.6.1
- "Material library: overview," Section 21.1.1
- *MOISTURE SWELLING
- "Defining moisture swelling" in "Defining a fluid-filled porous material," Section 12.12.3 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

Moisture swelling:

- defines the saturation-driven volumetric swelling of the solid skeleton of a porous medium in partially saturated flow conditions;
- can be used in the analysis of coupled wetting liquid flow and porous medium stress (see "Coupled pore fluid diffusion and stress analysis," Section 6.8.1); and
- can be either isotropic or anisotropic.

Moisture swelling model

The moisture swelling model assumes that the volumetric swelling of the porous medium's solid skeleton is a function of the saturation of the wetting liquid in partially saturated flow conditions. The porous medium is partially saturated when the pore liquid pressure, u_w, is negative (see "Effective stress principle for porous media," Section 2.8.1 of the Abaqus Theory Manual).

The swelling behavior is assumed to be reversible. The logarithmic measure of swelling strain is calculated with reference to the initial saturation so that

$$\varepsilon_{ii}^{ms} = r_{ii}\frac{1}{3}\left(\varepsilon^{ms}(s) - \varepsilon^{ms}(s^I)\right), \quad \text{(no sum on } i\text{)}$$

where $\varepsilon^{ms}(s)$ and $\varepsilon^{ms}(s^I)$ are the volumetric swelling strains at the current and initial saturations. A typical curve is shown in Figure 26.6.6–1. The ratios r_{11}, r_{22}, and r_{33} allow for anisotropic swelling as discussed below.

Defining volumetric swelling strain

Define the volumetric swelling strain, ε^{ms}, as a tabular function of the wetting liquid saturation, s. The swelling strain must be defined for the range $0.0 \le s \le 1.0$.

Input File Usage: *MOISTURE SWELLING

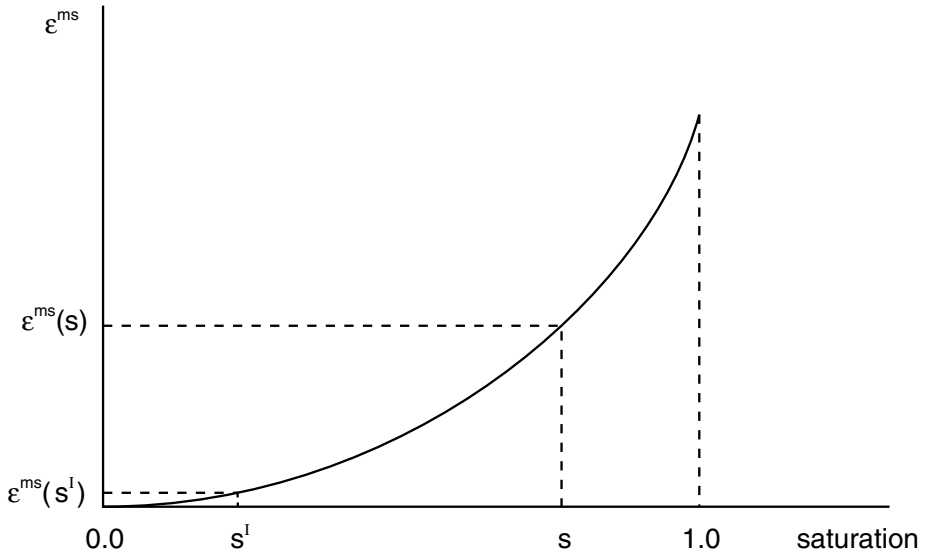


Figure 26.6.6–1 Typical volumetric moisture swelling versus saturation curve.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Moisture Swelling**

Defining initial saturation values

You can define the initial saturation values as initial conditions. If no initial saturation values are given, the default is fully saturated conditions (saturation of 1.0). For partial saturation the initial saturation and pore fluid pressure must be consistent, in the sense that the pore fluid pressure must lie within the absorption and exsorption values for the initial saturation value (see "Permeability," Section 26.6.2). If this is not the case, Abaqus/Standard will adjust the saturation value as needed to satisfy this requirement.

Input File Usage: *INITIAL CONDITIONS, TYPE=SATURATION

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Saturation** for the **Types for Selected Step**

Defining anisotropic swelling

Anisotropy can be included in moisture swelling behavior by defining the ratios r_{11}, r_{22}, and r_{33}, such that two or more of the three ratios differ. If the ratios r_{ii} are not specified, Abaqus/Standard assumes that the swelling is isotropic and that $r_{11} = r_{22} = r_{33} = 1.0$. The orientation of the moisture swelling strain directions depends on the user-specified local orientation (see "Orientations," Section 2.2.5).

Input File Usage: Use both of the following options:

*MOISTURE SWELLING
*RATIOS

The *RATIOS option should immediately follow the *MOISTURE SWELLING option.

Abaqus/CAE Usage: Property module: material editor: **Other→Pore Fluid→Moisture Swelling**: **Suboptions→Ratios**

Elements

The moisture swelling model can be used only in elements that allow for pore pressure (see "Choosing the appropriate element for an analysis type," Section 27.1.3).

26.7 User materials

- “User-defined mechanical material behavior,” Section 26.7.1
- “User-defined thermal material behavior,” Section 26.7.2

26.7.1 USER-DEFINED MECHANICAL MATERIAL BEHAVIOR

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References

- "UMAT," Section 1.1.40 of the Abaqus User Subroutines Reference Manual
- "VUMAT," Section 1.2.17 of the Abaqus User Subroutines Reference Manual
- *USER MATERIAL
- *DEPVAR
- "Specifying solution-dependent state variables," Section 12.8.2 of the Abaqus/CAE User's Manual, in the online HTML version of this manual
- "Defining constants for a user material," Section 12.8.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

User-defined mechanical material behavior in Abaqus:

- is provided by means of an interface whereby any mechanical constitutive model can be added to the library;
- requires that a constitutive model (or a library of models) is programmed in user subroutine **UMAT** (Abaqus/Standard) or **VUMAT** (Abaqus/Explicit); and
- requires considerable effort and expertise: the feature is very general and powerful, but its use is not a routine exercise.

Stress components and strain increments

The subroutine interface has been implemented using Cauchy stress components ("true" stress). For soils problems "stress" should be interpreted as effective stress. The strain increments are defined by the symmetric part of the displacement increment gradient (equivalent to the time integral of the symmetric part of the velocity gradient).

The orientation of the stress and strain components in user subroutine **UMAT** depends on the use of local orientations ("Orientations," Section 2.2.5).

In user subroutine **VUMAT** all strain measures are calculated with respect to the midincrement configuration. All tensor quantities are defined in the corotational coordinate system that rotates with the material point. To illustrate what this means in terms of stresses, consider the bar shown in Figure 26.7.1–1, which is stretched and rotated from its original configuration, AB, to its new position, $A'B'$. This deformation can be obtained in two stages; the bar is first stretched, as shown in Figure 26.7.1–2, and is then rotated by applying a rigid body rotation to it, as shown in Figure 26.7.1–3. The stress in the bar after it has been stretched is σ_{11}, and this stress does not change during the rigid body rotation. The $X'Y'$ coordinate system that rotates as a result of the rigid body rotation is the

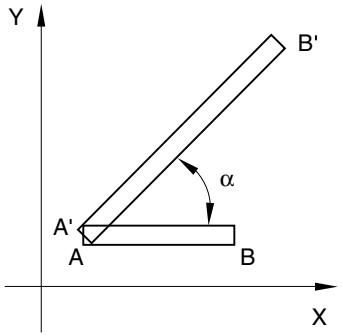


Figure 26.7.1–1 Stretched and rotated bar.

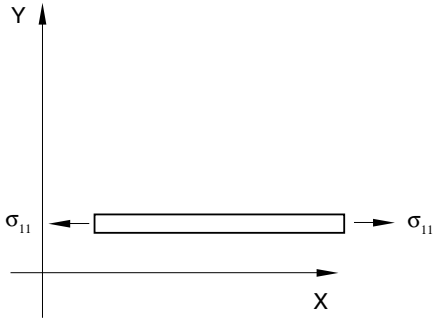


Figure 26.7.1–2 Stretching of bar.

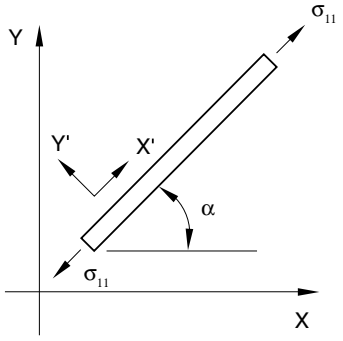


Figure 26.7.1–3 Rigid body rotation of bar.

corotational coordinate system. The stress tensor and state variables are, therefore, computed directly and updated in user subroutine **VUMAT** using the strain tensor since all of these quantities are in the

corotational system; these quantities do not have to be rotated by the user subroutine as is sometimes required in user subroutine **UMAT**.

The elastic response predicted by a rate-form constitutive law depends on the objective stress rate employed. For example, the Green-Naghdi stress rate is used in **VUMAT**. However, the stress rate used for built-in material models may differ. For example, most material models used with solid (continuum) elements in Abaqus/Explicit employ the Jaumann stress rate. This difference in the formulation will cause significant differences in the results only if finite rotation of a material point is accompanied by finite shear. For a discussion of the objective stress rates used in Abaqus, see "Stress rates," Section 1.5.3 of the Abaqus Theory Manual.

Material constants

Any material constants that are needed in user subroutine **UMAT** or **VUMAT** must be specified as part of a user-defined material behavior definition. Any other mechanical material behaviors included in the same material definition (except thermal expansion and, in Abaqus/Explicit, density) will be ignored; the user-defined material behavior requires that all mechanical material behavior calculations be programmed in subroutine **UMAT** or **VUMAT**. In Abaqus/Explicit the density ("Density," Section 21.2.1) is required when using a user-defined material behavior.

Input File Usage: In Abaqus/Standard use the following option to specify a user-defined material behavior:

∗USER MATERIAL, TYPE=MECHANICAL,
CONSTANTS=*number_of_constants*

In Abaqus/Explicit use both of the following options to specify a user-defined material behavior:

∗USER MATERIAL, CONSTANTS=*number_of_constants*
∗DENSITY

In either case you must specify the number of material constants being entered.

Abaqus/CAE Usage: In Abaqus/Standard use the following option to specify a user-defined material behavior:

Property module: material editor: **General→User Material**:
User material type: Mechanical

In Abaqus/Explicit use both of the following options to specify a user-defined material behavior:

Property module: material editor:
General→User Material: **User material type: Mechanical**
General→Density

Unsymmetric equation solver in Abaqus/Standard

If the user material's Jacobian matrix, $\partial \triangle \boldsymbol{\sigma} / \partial \triangle \boldsymbol{\varepsilon}$, is not symmetric, the unsymmetric equation solution capability in Abaqus/Standard should be invoked (see "Defining an analysis," Section 6.1.2).

Input File Usage: *USER MATERIAL, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*, UNSYMM

Abaqus/CAE Usage: Property module: material editor: **General→User Material**: **User material type: Mechanical**, toggle on **Use unsymmetric material stiffness matrix**

Material state

Many mechanical constitutive models require the storage of solution-dependent state variables (plastic strains, "back stress," saturation values, etc. in rate constitutive forms or historical data for theories written in integral form). You should allocate storage for these variables in the associated material definition (see "Allocating space" in "User subroutines: overview," Section 18.1.1). There is no restriction on the number of state variables associated with a user-defined material.

The user material subroutines are provided with the material state at the start of each increment, as described below. They must return values for the new stresses and the new internal state variables. State variables associated with **UMAT** and **VUMAT** can be output to the output database file (**.odb**) and results file (**.fil**) using the output identifiers SDV and SDV*n* (see "Abaqus/Standard output variable identifiers," Section 4.2.1, and "Abaqus/Explicit output variable identifiers," Section 4.2.2).

Material state in Abaqus/Standard

User subroutine **UMAT** is called for each material point at each iteration of every increment. It is provided with the material state at the start of the increment (stress, solution-dependent state variables, temperature, and any predefined field variables) and with the increments in temperature, predefined state variables, strain, and time.

In addition to updating the stresses and the solution-dependent state variables to their values at the end of the increment, subroutine **UMAT** must also provide the material Jacobian matrix, $\partial \triangle \boldsymbol{\sigma} / \partial \triangle \boldsymbol{\varepsilon}$, for the mechanical constitutive model. This matrix will also depend on the integration scheme used if the constitutive model is in rate form and is integrated numerically in the subroutine. For any nontrivial constitutive model these will be challenging tasks. For example, the accuracy with which the Jacobian matrix is defined will usually be a major determinant of the convergence rate of the solution and, therefore, will have a strong influence on computational efficiency.

Material state in Abaqus/Explicit

User subroutine **VUMAT** is called for blocks of material points at each increment. When the subroutine is called, it is provided with the state at the start of the increment (stress, solution-dependent state variables). It is also provided with the stretches and rotations at the beginning and the end of the increment. The **VUMAT** user material interface passes a block of material points to the subroutine on each call, which allows vectorization of the material subroutine.

The temperature is provided to user subroutine **VUMAT** at the start and the end of the increment. The temperature is passed in as information only and cannot be modified, even in a fully coupled thermal-stress analysis. However, if the inelastic heat fraction is defined in conjunction with the specific heat and conductivity in a fully coupled thermal-stress analysis in Abaqus/Explicit, the heat flux due to inelastic energy dissipation will be calculated automatically. If the **VUMAT** user subroutine is used to define an

adiabatic material behavior (conversion of plastic work to heat) in an explicit dynamics procedure, you must specify both the inelastic heat fraction and the specific heat for the material, and you must store the temperatures and integrate them as user-defined state variables. Most often the temperatures are provided by specifying initial conditions ("Initial conditions in Abaqus/Standard and Abaqus/Explicit," Section 33.2.1) and are constant throughout the analysis.

Deleting elements from an Abaqus/Explicit mesh using state variables

Element deletion in a mesh can be controlled during the course of an Abaqus/Explicit analysis through user subroutine **VUMAT**. Deleted elements have no ability to carry stresses and, therefore, have no contribution to the stiffness of the model. You specify the state variable number controlling the element deletion flag. For example, specifying a state variable number of 4 indicates that the fourth state variable is the deletion flag in **VUMAT**. The deletion state variable should be set to a value of one or zero in **VUMAT**. A value of one indicates that the material point is active, while a value of zero indicates that Abaqus/Explicit should delete the material point from the model by setting the stresses to zero. The structure of the block of material points passed to user subroutine **VUMAT** remains unchanged during the analysis; deleted material points are not removed from the block. Abaqus/Explicit will pass zero stresses and strain increments for all deleted material points. Once a material point has been flagged as deleted, it cannot be reactivated. An element will be deleted from the mesh only after all of the material points in the element are deleted. The status of an element can be determined by requesting output of the variable STATUS. This variable is equal to one if the element is active and equal to zero if the element is deleted.

Input File Usage: *DEPVAR, DELETE=*variable number*

Abaqus/CAE Usage: Property module: material editor: **General→Depvar**: **Variable number controlling element deletion:** *variable number*

Hourglass control and transverse shear stiffness

Normally the default hourglass control stiffness for reduced-integration elements in Abaqus/Standard and the transverse shear stiffness for shell, pipe, and beam elements are defined based on the elasticity associated with the material ("Section controls," Section 27.1.4; "Shell section behavior," Section 29.6.4; and "Choosing a beam element," Section 29.3.3). These stiffnesses are based on a typical value of the initial shear modulus of the material, which may, for example, be given as part of an elastic material behavior ("Linear elastic behavior," Section 22.2.1) included in the material definition. However, the shear modulus is not available during the preprocessing stage of input for materials defined with user subroutine **UMAT** or **VUMAT**. Therefore, you must provide the hourglass stiffness parameters (see "Methods for suppressing hourglass modes" in "Section controls," Section 27.1.4) when using **UMAT** to define the material behavior of elements with hourglassing modes; and you must specify the transverse shear stiffness (see "Choosing a beam element," Section 29.3.3, or "Shell section behavior," Section 29.6.4) when using **UMAT** or **VUMAT** to define the material behavior of beams and shells with transverse shear flexibility.

Use of UMAT with other subroutines

Various utility subroutines are also available in Abaqus/Standard for use with subroutine **UMAT**. These utility subroutines are discussed in "Obtaining stress invariants, principal stress/strain values and directions, and rotating tensors in an Abaqus/Standard analysis," Section 2.1.11 of the Abaqus User Subroutines Reference Manual.

User subroutine **UMATHT** can be used in conjunction with **UMAT** to define the constitutive thermal behavior of the material. The solution-dependent variables allocated in the material definition are accessible in both **UMAT** and **UMATHT**. In addition, user subroutines **FRIC**, **GAPCON**, and **GAPELECTR** are available for defining mechanical, thermal, and electrical interactions between surfaces.

Use with other material models

A number of material behaviors can be used in the definition of a material when its mechanical behavior is defined by user subroutine **UMAT** or **VUMAT**. These behaviors include density, thermal expansion, permeability, and heat transfer properties. Thermal expansion can alternatively be an integral part of the constitutive model implemented in **UMAT** or **VUMAT**.

The temperature available in **UMAT** is always the interpolated temperature field at the element integration points. Naturally, if the thermal expansion behavior is implemented in **UMAT**, it is defined in terms of the integration point temperature. When the temperature field is interpolated differently within an element compared to the displacement field in Abaqus/Standard, implementing the thermal expansion behavior in **UMAT** may lead to differences compared to the built-in thermal expansion behavior. This situation commonly arises for coupled temperature-displacement elements. For example, for first-order coupled temperature-displacement elements, the built-in thermal expansion behavior uses a constant temperature field over the whole element (see "Fully coupled thermal-stress analysis," Section 6.5.3), while the behavior in **UMAT** will be defined in terms of a linear temperature field.

For a material defined by user subroutine **UMAT** or **VUMAT**, mass proportional damping can be included separately (see "Material damping," Section 26.1.1), but stiffness proportional damping must be defined in the user subroutine by the Jacobian (Abaqus/Standard only) and stress definitions. Stiffness proportional damping cannot be specified if the user material is used in the direct steady-state dynamics procedure.

Elements

User subroutines **UMAT** and **VUMAT** can be used with all elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

26.7.2 USER-DEFINED THERMAL MATERIAL BEHAVIOR

Products: Abaqus/Standard Abaqus/CAE

References

- "UMATHT," Section 1.1.41 of the Abaqus User Subroutines Reference Manual
- *USER MATERIAL
- *DEPVAR
- "Defining constants for a user material," Section 12.8.4 of the Abaqus/CAE User's Manual, in the online HTML version of this manual

Overview

User-defined thermal material behavior in Abaqus/Standard:

- is provided by means of an interface whereby any thermal constitutive model can be added to the library;
- requires that a constitutive model (or a library of models) is programmed in user subroutine **UMATHT**; and
- requires considerable effort and expertise: the feature is very general and powerful, but its use is not a routine exercise.

Material constants

Any material constants that are needed in user subroutine **UMATHT** must be specified as part of a user-defined thermal material behavior definition. Any other thermal material behaviors included in the same material definition will be ignored: the user-defined thermal material behavior requires that all thermal behavior calculations are programmed in user subroutine **UMATHT**.

Input File Usage: *USER MATERIAL, TYPE=THERMAL, CONSTANTS=*number_of_constants*

You must specify the number of constants being entered.

Abaqus/CAE Usage: Property module: material editor: **General→User Material**: **User material type: Thermal**

Unsymmetric equation solver

When the conductivity is defined in user subroutine **UMATHT** as a strong function of temperature, the heat transfer equilibrium equations become nonsymmetric and you may choose to invoke the unsymmetric equation solution capability; otherwise, convergence may be poor.

Input File Usage: *USER MATERIAL, TYPE=THERMAL, CONSTANTS=*number_of_constants*, UNSYMM

Abaqus/CAE Usage: Property module: material editor: **General→User Material**: **User material type: Thermal**, toggle on **Use unsymmetric material stiffness matrix**

Material state

Many thermal constitutive models require the storage of solution-dependent state variables. These state variables might include microstructure or phase content information when the material undergoes phase changes. You should allocate storage for these variables in the associated material definition (see "Allocating space" in "User subroutines: overview," Section 18.1.1). There is no restriction on the number of state variables associated with a user-defined material.

User subroutine **UMATHT** is called for each material point at each iteration of every increment. It is provided with the thermal state of the material at the start of the increment (solution-dependent state variables, temperature, and any predefined field variables) and with the increments in temperature, predefined state variables, and time.

Required calculations

Subroutine **UMATHT** must perform the following functions: it must define the internal energy per unit mass and its variation with temperature and spatial gradients of temperature; it must define the heat flux vector and its variation with respect to temperature and spatial gradients of temperature; and it must update the solution-dependent state variables to their values at the end of the increment. The components of the heat flux and spatial gradients in user subroutine **UMATHT** are in directions that depend on the use of local orientations (see "Orientations," Section 2.2.5).

Use with other user subroutines

User subroutine **UMAT** can be used in conjunction with **UMATHT** to define the constitutive mechanical behavior of the material. The solution-dependent variables allocated in the material definition are accessible in both **UMATHT** and **UMAT**. In addition, user subroutines **FRIC**, **GAPCON**, and **GAPELECTR** are available for defining mechanical, thermal, and electrical interactions between surfaces.

Use with other material models

Density, mechanical properties, and electrical properties can be included in the definition of a material whose constitutive thermal behavior is defined by user subroutine **UMATHT**.

Elements

User subroutine **UMATHT** can be used with all elements in Abaqus/Standard that include thermal behavior (elements with temperature degrees of freedom such as pure heat transfer, coupled thermal-stress, and coupled thermal-electrical elements).

About SIMULIA

SIMULIA is the Dassault Systèmes brand that delivers a scalable portfolio of Realistic Simulation solutions including the Abaqus product suite for Unified Finite Element Analysis; multiphysics solutions for insight into challenging engineering problems; and lifecycle management solutions for managing simulation data, processes, and intellectual property. By building on established technology, respected quality, and superior customer service, SIMULIA makes realistic simulation an integral business practice that improves product performance, reduces physical prototypes, and drives innovation. Headquartered in Providence, RI, USA, with R&D centers in Providence and in Vélizy, France, SIMULIA provides sales, services, and support through a global network of regional offices and distributors. For more information, visit www.simulia.com.

About Dassault Systèmes

As a world leader in 3D and Product Lifecycle Management (PLM) solutions, Dassault Systèmes brings value to more than 100,000 customers in 80 countries. A pioneer in the 3D software market since 1981, Dassault Systèmes develops and markets PLM application software and services that support industrial processes and provide a 3D vision of the entire lifecycle of products from conception to maintenance to recycling. The Dassault Systèmes portfolio consists of CATIA for designing the virtual product, SolidWorks for 3D mechanical design, DELMIA for virtual production, SIMULIA for virtual testing, ENOVIA for global collaborative lifecycle management, and 3DVIA for online 3D lifelike experiences. Dassault Systèmes' shares are listed on Euronext Paris (#13065, DSY.PA), and Dassault Systèmes' ADRs may be traded on the US Over-The-Counter (OTC) market (DASTY). For more information, visit www.3ds.com.

www.3ds.com Abaqus, the 3DS logo, SIMULIA, CATIA, SolidWorks, DELMIA, ENOVIA, 3DVIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries. Other company, product, and service names may be trademarks or service marks of their respective owners. © Dassault Systèmes, 2012

DASSAULT SYSTEMES